

FORM PTO-1390 (Modified)
(REV 10-95)

U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE

ATTORNEY'S DOCKET NUMBER

TRANSMITTAL LETTER TO THE UNITED STATES

MAT-V07839

DESIGNATED/ELECTED OFFICE (DO/EO/US)

U.S. APPLICATION NO. (IF KNOWN, SEE 37 CFR

CONCERNING A FILING UNDER 35 U.S.C. 371

09/403560

INTERNATIONAL APPLICATION NO.
PCT/JP98/01837INTERNATIONAL FILING DATE
22 April 1998 (22.04.98)PRIORITY DATE CLAIMED
24 April 1997 (24.04.97)

TITLE OF INVENTION

DATA TRANSFER METHOD

APPLICANT(S) FOR DO/EO/US

Takuya NISHIMURA, Hiroyuki IITSUKA and Masazumi YAMADA

Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:

1. ☒ This is a **FIRST** submission of items concerning a filing under 35 U.S.C. 371.
2. ☐ This is a **SECOND** or **SUBSEQUENT** submission of items concerning a filing under 35 U.S.C. 371.
3. ☒ This is an express request to begin national examination procedures (35 U.S.C. 371(f)) at any time rather than delay examination until the expiration of the applicable time limit set in 35 U.S.C. 371(b) and PCT Articles 22 and 39(1).
4. ☐ A proper Demand for International Preliminary Examination was made by the 19th month from the earliest claimed priority date.
5. ☒ A copy of the International Application as filed (35 U.S.C. 371 (c) (2))
 - a. ☒ is transmitted herewith (required only if not transmitted by the International Bureau).
 - b. ☐ has been transmitted by the International Bureau.
 - c. ☐ is not required, as the application was filed in the United States Receiving Office (RO/US).
6. ☒ A translation of the International Application into English (35 U.S.C. 371(c)(2)).
7. ☒ A copy of the International Search Report (PCT/ISA/210).
8. ☐ Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. 371 (c)(3))
 - a. ☐ are transmitted herewith (required only if not transmitted by the International Bureau).
 - b. ☐ have been transmitted by the International Bureau.
 - c. ☐ have not been made; however, the time limit for making such amendments has NOT expired.
 - d. ☐ have not been made and will not be made.
9. ☐ A translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371(c)(3)).
10. ☒ An oath or declaration of the inventor(s) (35 U.S.C. 371 (c)(4)). (unexecuted)
11. ☒ A copy of the International Preliminary Examination Report (PCT/IPEA/409).
12. ☐ A translation of the annexes to the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371 (c)(5)).

Items 13 to 18 below concern document(s) or information included:

13. ☒ An Information Disclosure Statement under 37 CFR 1.97 and 1.98.
14. ☐ An assignment document for recording. A separate cover sheet in compliance with 37 CFR 3.28 and 3.31 is included.
15. ☒ A **FIRST** preliminary amendment.
A **SECOND** or **SUBSEQUENT** preliminary amendment.
16. ☒ A substitute specification.
17. ☐ A change of power of attorney and/or address letter.
18. ☒ Certificate of Mailing by Express Mail
19. ☐ Other items or information:

Marked-up copy of specification showing additions and deletions.

U.S. APPLICATION NO. (IF KNOWN, SEE 37 CFR 09/403560)	INTERNATIONAL APPLICATION NO. PCT/JP98/01837	ATTORNEY'S DOCKET NUMBER MAT-V07839
--	---	--

20. The following fees are submitted:

BASIC NATIONAL FEE (37 CFR 1.492 (a) (1) - (5)) :

- ☒ Search Report has been prepared by the EPO or JPO **\$840.00**
- ☐ International preliminary examination fee paid to USPTO (37 CFR 1.482) **\$670.00**
- ☐ No international preliminary examination fee paid to USPTO (37 CFR 1.482) but international search fee paid to USPTO (37 CFR 1.445(a)(2)) **\$760.00**
- ☐ Neither international preliminary examination fee (37 CFR 1.482) nor international search fee (37 CFR 1.445(a)(2)) paid to USPTO **\$970.00**
- ☐ International preliminary examination fee paid to USPTO (37 CFR 1.482) and all claims satisfied provisions of PCT Article 33(2)-(4) **\$96.00**

ENTER APPROPRIATE BASIC FEE AMOUNT =**CALCULATIONS PTO USE ONLY****\$840.00**

Surcharge of **\$130.00** for furnishing the oath or declaration later than ☐ 20 ☐ 30 months from the earliest claimed priority date (37 CFR 1.492 (e)).

\$0.00

CLAIMS	NUMBER FILED	NUMBER EXTRA	RATE
Total claims	66 - 20 =	46	x \$18.00
Independent claims	1 - 3 =	0	x \$78.00

\$828.00**\$0.00**

Multiple Dependent Claims (check if applicable).

☒**\$260.00****TOTAL OF ABOVE CALCULATIONS =****\$1,928.00**

Reduction of 1/2 for filing by small entity, if applicable. Verified Small Entity Statement must also be filed (Note 37 CFR 1.9, 1.27, 1.28) (check if applicable).

☐**\$0.00****SUBTOTAL =****\$1,928.00**

Processing fee of **\$130.00** for furnishing the English translation later than ☐ 20 ☐ 30 months from the earliest claimed priority date (37 CFR 1.492 (f)).

+

\$0.00**TOTAL NATIONAL FEE =****\$1,928.00**

Fee for recording the enclosed assignment (37 CFR 1.21(h)). The assignment must be accompanied by an appropriate cover sheet (37 CFR 3.28, 3.31) (check if applicable).

☐**\$0.00****TOTAL FEES ENCLOSED =****\$1,928.00**

Amount to be:

refunded

\$

charged

\$

☒ A check in the amount of **\$1,928.00** to cover the above fees is enclosed.

☐ Please charge my Deposit Account No. _____ in the amount of _____ to cover the above fees.
A duplicate copy of this sheet is enclosed.

☒ The Commissioner is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. **18-0350** A duplicate copy of this sheet is enclosed.

NOTE: Where an appropriate time limit under 37 CFR 1.494 or 1.495 has not been met, a petition to revive (37 CFR 1.137(a) or (b)) must be filed and granted to restore the application to pending status.

SEND ALL CORRESPONDENCE TO:

Lawrence E. Ashery
Ratner & Prestia
P.O. Box 980
Valley Forge, PA 19482
(610) 407-0700

SIGNATURE

Lawrence E. Ashery

NAME

34,515

REGISTRATION NUMBER

October 25, 1999

DATE

DATA TRANSFER METHOD

FIELD OF THE INVENTION

5 The present invention relates to the field of digital data transfer methods, more particularly to the transfer of data in which normal digital data and encrypted digital data co-exist in the same data.

BACKGROUND OF THE INVENTION

10 One conventional data transfer method adopts the IEEE1394 standard (IEEE: The Institute of Electrical and Electronics Engineers, Inc.). (Reference: IEEE Std 1394: 1995, High Performance Serial Bus.) In data transfer specified by the IEEE 1394 standard, there are two methods of communication. One is isochronous communication, which is suitable for transferring synchronous data such as digital video signals and digital audio signals. The other is asynchronous
15 communication, which is suitable for transferring asynchronous data such as control signals. Both methods of communication are applicable on the IEEE 1394 bus network. Isochronous communication is what is called Broadcast communication, and an isochronous packet output from one device coupled to the IEEE 1394 bus is receivable by all the other devices coupled to the same bus. On
20 the other hand, asynchronous communication is applicable to both one-to-one communication and one-to-N broadcast communication. Each asynchronous

packet output from one device coupled to the bus contains an identifier specifying the device(s) to which that packet is addressed. If this identifier specifies a particular device, only the device specified by the identifier receives the asynchronous packet. If the identifier specifies broadcast, all the devices coupled to the same bus receive the asynchronous packet.

At present, the IEC (International Electrotechnical Commission) is preparing to stipulate the IEC1883 standard (hereafter referred to as AV protocol) for transferring digital audio signals and digital video signals or transmitting data between devices coupled to an IEEE 1394 bus, employing the data transfer method conforming to the IEEE 1394 standard. In the AV protocol, video and audio data is located in the isochronous packet as shown in Fig. 5 and transferred. The isochronous packet includes a CIP (Common Isochronous Packet) header. The CIP header carries information that includes the type of AV data, the identification number of the device which is sending the isochronous packet, and the like.

Fig. 5 shows the format of the isochronous packet used in the AV protocol. The isochronous packet comprises an isochronous packet header 900, header CRC 901, isochronous payload 902, and data CRC 903. The isochronous packet header 900 contains a tag 907. The tag 907 shows that the isochronous packet conforms to the AV protocol when its value is 1. When the value of the tag 907 is 1, which means that the isochronous packet conforms to the AV protocol, the isochronous payload 902 has a CIP header 904 at its beginning. The CIP header 904 comprises a source ID 906 which identifies the device transmitting the isochronous packet. The CIP header 904 also comprises FMT 908 and FDF 909 which specify the type of actual data 905 in the isochronous

payload 902. Digital AV data is contained in the actual data 905, but the actual data 905 is not always contained in the isochronous payload 902. Some packets may have an isochronous payload 902 which contains only the CIP header 904 without the actual data 905.

5 There is a group of commands called the AV/C Command Set for controlling devices in accordance with the AV protocol (Reference: 1394 TRADE ASSOCIATION Specification for AV/C Digital Interface Command Set Version 1.0, September 13, 1996). These commands and their responses are transferred by means of asynchronous communication.

10 In the conventional data transfer method as described above, compatibility with conventional devices which are not designed for transferring an encrypted isochronous payload 902 cannot be secured when an encrypted isochronous packet, which contains the isochronous payload 902 which has been encrypted for copyright protection, is sent. More specifically, conventional
15 devices are designed with the precondition that the CIP header 904 is normally positioned at the beginning of the isochronous payload 902. Accordingly, if the isochronous payload 902 is encrypted, conventional devices cannot correctly read out the encrypted CIP header 904, and decide that the isochronous packet does not conform to the AV protocol. A device receiving encrypted isochronous
20 packets thus may not operate properly. In other words, such receiving devices cannot determine the type of data contained in the actual data 905, resulting in an inability to identify the device transmitting the isochronous packet. In addition, asynchronous communication such as queries to the sending device are disabled. Accordingly, normal receiving operations cannot be carried out.

Furthermore, if the isochronous packet output from the sending device is encrypted while the receiving device is receiving the data, some conventional devices may not be able to correctly read out the CIP header 904 as soon as encryption starts, resulting in inability to receive data properly.

5 In order to send AV information encrypted for copyright protection from the sending device and decrypt the encrypted AV data by the authorized receiving device, the sending device needs to give decrypting information for decryption to the authorized receiving device. In the conventional data transfer method, however, the sending device may be required to execute extremely
10 complicated procedures in order to specify the receiving device. More specifically, each isochronous packet contains the source ID 906 which is the identifier of the sending device, but these packets do not contain information that identifies which device is authorized to receive these packets. The sending device thus cannot check which device is receiving the isochronous packets during
15 transmission of the isochronous packets. In order to find which of the devices coupled to the IEEE 1394 bus is receiving the data, the sending device may require to query the data receiving status of every device coupled to the same bus. This makes the procedures for giving key information for decryption extremely complicated.

SUMMARY OF THE INVENTION

20 A data transfer method of the present invention satisfies the conventional communication standard even in the case of sending encrypted video and audio information via isochronous communication. In addition, the
25 present invention offers a data transfer method which prevents erroneous

operation even if conventional receiving devices receive isochronous packets containing encrypted video and audio data.

The present invention still further offers a data transfer method which significantly simplifies procedures for giving key information for decryption from a sending device to an authorized receiving device.

In a data transfer method of the present invention, synchronous data transferred via isochronous communication contains i) encryption identification information which indicates encryption status of actual data and ii) actual data, and only the actual data is encrypted.

To solve another problem in the conventional data transfer method, the encryption identification information which indicates encryption status of the actual data in the synchronous data is sent together with the actual data from the sending device, so that receiving device can detect that the actual data is encrypted based on this encryption identification information and requests decrypting information from the sending device in the data transfer method of the present invention. Then, the receiving device receiving the decrypting information sent from the sending device upon request decrypts the actual data using this decrypting information to complete data transfer.

Also in the data transfer method of the present invention, the receiving device receiving synchronous data checks for the encryption identification information contained in the synchronous data. If the receiving device detects that the actual data is encrypted, the receiving device requests decrypting information for decrypting the actual data from the sending device. This request is made using a command in the AV/C set via asynchronous communication. At receiving this request, the sending device checks the packet

header of received command to identify the device making the request, i.e., the receiving device. The sending device then gives decrypting information to the identified receiving device using a command via asynchronous communication, enabling to realize the data transfer method with extremely simple procedures for giving decrypting information from the sending device to the receiving device.

Moreover, in the data transfer method of the present invention, only the actual data in the synchronous data is encrypted, and the encryption identification information indicating the encryption status of the actual data is included in the synchronous data. This enables to transfer the CIP header without being encrypted, preventing erroneous operation when the conventional device receives such encrypted synchronous data. In other words, the present invention realizes a data transfer method which assures compatibility with the conventional data transfer method and eliminates the possibility of erroneous operation when the conventional receiving device receives encrypted synchronous data.

Furthermore, the data transfer method of the present invention eliminates the possibility of erroneous operation of the receiving device receiving data when encryption of synchronous data starts while continuously receiving synchronous data from the sending device because the CIP header is not encrypted and transferred as it is.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a schematic view of a format of a CIP header in accordance with a preferred embodiment of the present invention.

Fig. 2 is a block diagram illustrating functions of sending and receiving devices in accordance with the preferred embodiment of the present invention.

Fig. 3A is a format of AKE status command in accordance with the preferred embodiment of the present invention.

Fig. 3B is a format of AKE response to the AKE status command in accordance with the preferred embodiment of the present invention.

Fig. 3C is a format of AKE control command in accordance with the preferred embodiment of the present invention.

Fig. 4 is a schematic view illustrating procedures for transmitting an asynchronous packet between sending and receiving devices in accordance with the preferred embodiment of the present invention.

Fig. 5 is a format of isochronous packet in a data transfer method of the prior art.

DESCRIPTION OF THE PREFERRED EMBODIMENT

A preferred embodiment of the present invention is described next with reference to the drawings.

Fig. 1 shows a format of the payload of an isochronous packet to be transferred in the preferred embodiment of the present invention. The preferred embodiment is one example of the transfer of a TSP (Transport Packet) in accordance with MPEG (the Moving Picture Expert Group) specifications. The ENC (hereafter referred to as encryption information) 910 indicates whether the actual data 905 is encrypted or not.

Fig. 2 shows the relation between sending and receiving devices in the preferred embodiment of the present invention. A sending device 110 and receiving device 128 are coupled via an IEEE 1394 bus (hereafter referred to as a 1394 bus) 111.

5 First, the functions of each block in the sending device 110 are described.

A signal source 100 outputs an MPEG transport packet TSP (not illustrated) in an 188 byte unit, which will be sent via the 1394 bus 111, to an encrypter 101. In other words, in the preferred embodiment, the signal source
10 100 outputs data with a fixed length of 188 bytes. The encrypter 101 encrypts and outputs the TSP received from the signal source 100 using an encryption key 109 provided by a key generator 106. In the preferred embodiment, the encryption key 109 is equivalent to the decrypting information. An output command 105 is a command from the key generator 106 to the encrypter 101.
15 There are three types of commands: normal output, encrypted output, and empty output. If the encrypter 101 receives the output command 105 for normal output, the TSP received from the signal source 100 is output without modification, and registers the value 0 as the encrypting information 910. If the output command 105 is for encrypted output, the encrypter 101 encrypts the TSP with the
20 encryption key 109 received from the key generator 106, and registers the value 1 as the encrypting information 910. If the output command 105 is for empty output, the encrypter 101 outputs an empty signal (not illustrated) every time it receives a TSP from the signal source 100, and registers the value 1 as the encrypting information 910. A source packet generator 102 adds a 4-byte source
25 packet header to the 188-byte TSP received from the encrypter 101, and outputs

a 192-byte source packet (actual data 905). A CIP block generator 103 adds a CIP header 954 to the source packet received from the source packet generator 102, and outputs an isochronous payload 952. Here, the CIP block generator 103 places the encrypting information 910 received from the encrypter 101 in the CIP header 954. An isochronous packet generator 107 adds an isochronous packet header 900, header CRC 901, and data CRC 903 to the isochronous payload 952 received from the CIP block generator 103, and outputs an isochronous packet. Since the content of the isochronous payload 952 is data that conforms to the AV protocol, the value of the tag 907 is set to 1. The key generator 106 sends the encryption key 109 to the receiving device 128 by communicating the asynchronous packet with the receiving device 128, as shown in Fig. 3, which is described later. The key generator 106 also outputs the encryption key 109 to the encrypter 101 as described above.

A 1394 packet I/O controller 108 inputs and outputs isochronous and asynchronous packets between the 1394 bus 111 and sending device 110. More specifically, the 1394 packet I/O controller 108 outputs the isochronous packet received from the isochronous packet generator 107 and asynchronous packet received from the key generator 106 to the 1394 bus 111, and also outputs asynchronous packet received from the 1394 bus 111 to the key generator 106.

Next, functions of each block of the receiving device 128 are described.

A 1394 packet I/O controller 127 inputs and outputs isochronous and asynchronous packets between the 1394 bus 111 and receiving device 128. More specifically, the 1394 packet I/O controller 127 outputs the isochronous packet received from the 1394 bus 111 to a payload extractor 123, and outputs

asynchronous packet received from the 1394 bus 111 to a key generator 125. The 1394 packet I/O controller 127 also outputs asynchronous packet received from the key generator 125 to the 1394 bus 111.

The payload extractor 123 receives the isochronous packet, transmitted from the 1394 bus 111, from the 1394 packet I/O controller 127. When the value of the isochronous packet tag 907 is 1, the payload extractor 123 determines that an isochronous payload 952 contains data conforming to the AV protocol, and outputs the isochronous payload 952 to an actual data extractor 122. When received isochronous payload 952 contains the actual data 905, the actual data extractor 122 outputs the actual data 905 to a decrypter 121, after removing the CIP header 954 placed at the beginning of the isochronous payload 952. The actual data extractor 122 also outputs the source ID 906 and encrypting information 910 extracted from the CIP header 954 to the key generator 125. The encrypting information 910 is also output to the decrypter 121. The key generator 125 receives an encryption key 126 as a result of exchanging asynchronous packet with the sending device 110 via asynchronous communication, which is described later, and outputs the encryption key 126 to the decrypter 121. When the value of the encrypting information 910 received from the actual data extractor 122 is 0, the decrypter 121 outputs the actual data 905 received from the actual data extractor 122 to an AV generator 120 as it is. When the value of the encrypting information 910 is 1, the decrypter 121 decrypts the actual data 905 using the encryption key 126 received from the key generator 125, and outputs decrypted actual data 905 to the AV generator 120.

Next, the transmission of an asynchronous packet via the aforementioned asynchronous communication setup is described.

Figs. 3A to 3C illustrate how the format of the asynchronous packet is transmitted by asynchronous communication. More specifically, Figs. 3A and 3C show the command formats of the AKE commands (AKE: Authentication and Key Exchange) communicated between the key generators 106 and 125. Fig. 3B shows the response format. These commands and responses belong to the AV/C Command Set, and are communicated between the sending device 110 and receiving device 128 using the asynchronous communication. By communicating these commands and responses, the sending device 110 and receiving device 128 exchange information required for the authentication of each other and encryption keys 109 and 126. The AKE commands comprise AKE control commands for requesting a target device to carry out a specific operation, and an AKE status commands for querying the status and capabilities of the target device.

Fig. 3A shows the format of the AKE status command. In the AKE status command, an operation code 208 indicates that this command is an AKE command. The value of the algorithm ID 200 is set at 0, with other values reserved for future extension.

Fig. 3B shows the format of responses to the AKE status commands. This is a response sent back from the device receiving the AKE status command to the device issuing the AKE status command. There are multiple procedures for exchanging information for mutual authentication and transmission of encryption keys 109 and 126 between the sending device 110 and receiving device 128. In an algorithm field 201, the identifier for an information exchange procedure which the device returning an applicable response can execute is assigned in bits.

In other words, the receiving device 128 exchanges several commands and

responses with the sending device 110 after an encrypted TSP is detected in line with the aforementioned procedures and before receiving the encryption keys 109 and 126. There is more than one information exchange procedure for communicating these commands and responses. The device sending back the response designates the executable information exchange procedure by setting 1 to an applicable bit in the algorithm field 201. Since the size of the algorithm field 201 is 16 bits, a maximum of 16 types of information exchange procedures can be indicated. The maximum data length 212 indicates the longest receivable data length in the form of bytes for exchanging AKE commands and responses.

Fig. 3C shows the format of the AKE control commands. The algorithm field 201 in the AKE control commands set informs of an executed information exchange procedure when the value of the algorithm ID 200 is 0. Only one bit in the algorithm field 201 of the AKE control command and the response to AKE control commands is set at 1, and the other bits are 0. A bit having the value 1 indicates the information exchange procedure being used. A label 202 is used for identifying correspondence between AKE control commands. For example, let's say a certain information exchange procedure specifies that the device receiving an AKE control command needs to return a different AKE control command corresponding to the AKE control command received when the AKE control command is sent from one device to another. In this case, the label 202 inserted in the returned AKE control command will have the same value as the label 202 inserted in the first AKE control command received, in order to clarify the correlation between both AKE control commands. In step No. 203, a serial number from 1 is given to each AKE

control command in the sequence of communication in the information exchange procedure.

A subfunction 299 takes the values shown in Table 1, and the meaning of each AKE command is determined by these values.

5

Table 1

Subfunction	Value
Make-response	0016
Verify-me	0116
Create-key-	1016
information Reconstruct-key	1116
Exchange	2016

If the subfunction 299 is the make-response, this AKE control command challenges the authentication of the device receiving this command. Here, the data 207 contains authentication challenge data expressed as random numbers to authenticate the receiving device. The device receiving this command returns an AKE control command whose subfunction 299 is set to verify-me.

10

When returning the AKE control command, the data stored in the data 207 is the authentication response data which is a result of a predetermined operation with respect to the authentication challenge data in the received data 207. The key information used for this operation is a key given only to an authorized device in advance. Whether the device executing the operation is an authorized device or not can be determined by checking the returned authentication response data.

15

If the subfunction 299 is the create-key-information, this AKE control command requests the encryption key 109 to the device receiving this

20

command. The device receiving this AKE control command returns the AKE control command whose subfunction 299 is set to reconstruct-key. At this point, the encrypted encryption key 109 is stored in the data 207 and returned.

If the subfunction 299 is the exchange, this AKE control command requests the exchange of key information between devices sending and receiving the command. This key information is stored in the data 207 and transferred for indirect authentication between devices or the creation of a common key.

Values of the subfunction other than those specified in Table 1 are reserved for future extension. The channel No. 204 indicates the channel number for isochronous communication between the sending device 110 and receiving device 128. This channel No. 204 is valid only when the subfunction 299 is set to the create-key-information or reconstruct-key. In other cases, this value will be set to FF in hexadecimal format. Block No. 205 and total block No. 206 are used when data which should be handled by the AKE control command cannot be sent by one AKE command. In this case, applicable data is divided into blocks, and transferred in several transmissions. The total block No. 206 indicates the number of divided blocks in applicable data. The block No. 205 indicates the number of each block in the data 207. The data length 209 indicates the valid data length, as bytes, in the data 207. The data 207 is data exchanged by the AKE control command. The device receiving the AKE control command returns a response to that specific AKE control command. The format and value of the response are the same as those of the received AKE control command. The only detail which differs is that the response does not contain the data 207.

Fig. 4 shows a time sequence example of AV/C commands which are exchanged between the sending device 110 and receiving device 128 before

sending the encryption keys 109 and 126 from the sending device 110 to receiving device 128. First, operations of both devices before exchanging AV/C commands shown in Fig. 4 are briefly described.

An initial condition is that non-encrypted TSP is sent from the sending device 110. The TSP output from the signal source 100 is input to the encrypter 101. Since the output command 105 is set to the normal output, the encrypter 101 outputs TSP as it is without encryption to the source packet generator 102, and registers the value 0 as the encrypting information 910. The source packet generator 102 adds 4-byte source packet header to the TSP received, and outputs it to the CIP block generator 103. The CIP block generator 103 adds 8-byte CIP header 954, and outputs it as isochronous payload 952 to the isochronous packet generator 107. Here, the encrypting information 910 contained in the CIP header 954 is 0 which is input from the encrypter 101. The isochronous packet generator 107 adds the isochronous packet header 900, header CRC 901, and data CRC 903 to the received isochronous payload 952 to create the isochronous packet. This isochronous packet is output to the 1394 bus 111 by the 1394 packet I/O controller 108. Since the applicable isochronous packet conforms to the AV protocol, the tag 907 in the isochronous packet header 900 is set to 1.

When the TSP output from the signal source 100 is changed, which means that AV information changes from that unprotected AV information to copy-protected AV information, the key generator 106 detects this change, and changes the output command 105 from the normal output to empty output. At the same time, the encryption key 109 for encrypting TSP is given to the encrypter 101.

When the output command 105 is for empty output, the encrypter 101 outputs an empty signal to the source packet generator 102 every time it receives a TSP from the signal source 100, and registers the value 1 as the encrypting information 910. At receiving the empty signal from the encrypter 101, the source packet generator 102 transmits the received empty signal as it is to the CIP block generator 103 without adding the source packet header. When the CIP block generator 103 receives the empty signal, it outputs only the CIP header 954 to the isochronous packet generator 107. Here, the encrypting information 910 in the CIP header 954 uses the value 1 output from the encrypter 101. The isochronous packet generator 107 creates an isochronous packet as the isochronous payload 952 using the CIP header 954 received from the CIP block generator 103, and outputs it to the 1394 packet I/O controller 108. Since this isochronous packet conforms to the AV protocol, the value of the tag 907 is set to 1. The 1394 packet I/O controller 108 outputs received isochronous packet to the 1394 bus 111. This isochronous packet is continuously output, and the isochronous packet only containing the CIP header 954 in this isochronous payload 952 is continuously output to the 1394 bus 111. The receiving device 128 receiving this isochronous packet checks its tag 907 by the 1394 packet I/O controller 127, detects that the isochronous packet conforms to the AV protocol, and then outputs this isochronous packet to the payload extractor 123. The payload extractor 123 extracts the isochronous payload 952 from received isochronous packet, and outputs it to the actual data extractor 122. The actual data extractor 122 outputs the encrypting information 910 and source ID 906 in the CIP header 954 to the key generator 125. After the key generator 125 detects that the value of the encrypting information 910 is 1, the key generator 125

learns from the source ID 906 that device outputting the isochronous packet is the sending device 110. Then, the key generator 125 finally goes onto a process for requesting the encryption keys 109 and 126 using the A/C commands, as shown in Fig. 4.

5 In Fig. 4, the AKE status command 300 is first sent from the receiving device 128 to sending device 110. This enables the receiving device 128 to query information exchange procedure that can be used by the sending device 110. Replying to this query, the sending device 110 returns the AKE response 301 to the receiving device 128. Information exchange procedure which
10 the sending device 110 can execute is assigned in bits in the algorithm field 201 of the AKE response 301. This allows the receiving device 128 to learn which information exchange procedures can be executed by the sending device 110. For example, if the sending device 110 can execute the second and sixth information exchange procedures, binary indication in the algorithm field 201 of the AKE
15 response 301 will be 0000000000100010.

The receiving device 128 receiving the AKE response 301 selects one optimal procedure from information exchange procedures that both sending device 110 and receiving device can execute. Then, A/C commands are exchanged according to the selected exchange procedure. Let's say the receiving
20 device 128 can execute the second and eighth information exchange procedures. Then, the information exchange procedure which can be executed by both sending device 110 and receiving device 128 is only the second procedure. Accordingly, the rest of authentication and information exchange are executed using the second procedure. In the AKE control command in this procedure, the
25 value of algorithm ID 200 will be 0 and the value of the algorithm field 201 will

be 00000000000000010 in binary indication. The information exchange procedure specifies not only the sequence of exchanging a range of AKE control commands but also a format and processing method of the data 207 sent by each AKE control command.

5 In accordance with the second information exchange procedure, the key generator 125 sends the make-response command 302 to the sending device 110. In the data 207 of this make-response command 302, two random numbers RRa and RRb generated by the key generator 125 are encrypted, and the algorithm field 201 contains identification information indicating the use of the
10 second procedure. The key used for encryption is a common secret key given to both authorized sending device and receiving device in advance. The key generator 106 receiving the make-response command 302 checks the algorithm field 201 of the received make-response command 302, and learns to use the second procedure for the rest of the authentication and information exchange.
15 Since the key generator 106 can execute the second procedure, the key generator 106 knows that the data 207 of the make-response command 302 sent in accordance with this second procedure contains two random numbers encrypted by this secret key. After taking out two random numbers RRa and RRb from the data 207 using this secret key, the key generator 106 returns a response 303 to
20 inform that a response can be generated. Then, the key generator 106 stores one of the random numbers RRa taken out in the data 207, and sends the verify-me command 304 to the receiving device 128. This is the response requested by the previous make-response command 302. Hereafter, the algorithm field 201 of each AKE command exchanged between the sending device 110 and receiving

device 128 always contain the identification information indicating the second procedure.

The key generator 125 receiving the verify-me command 304 confirms that RRa in the data 207 conforms to the random number RRa generated by itself, and then returns a response 305 to the verify-me command 304 to inform that verification has completed successfully. The key generator 125 then finally authenticates that the sending device 110 is an authorized sending device.

The sending device 110 then use the make-response command 306 and verify-me command 308 in accordance with the procedures after the make-response command 302 described above to confirm that the receiving device 128 is an authorized receiving device. However, the random number used here is RTa and RTb, and the random number sent back by the verify-me command 308 is RTb.

Now that both sending device 110 and receiving device 128 know the random numbers RRb and RTb, and have confirmed that both are authorized devices, the key generator 106 and key generator 125 separately generates a temporary key (not illustrated) from RRb and RTb using a common operation method specified by the second procedure. These temporary keys are a common key only between the sending device 110 and receiving device 128.

Next, the key generator 125 sends the create-key-information command 310 to the sending device 110. A channel number of the isochronous packet that the receiving device 128 is currently receiving is stored in the channel No. 204 of the create-key-information command 310. The key generator 106 receiving this create-key-information command 310 encrypts the encryption key

109 to be used for encrypting TSP with the aforementioned temporary key, and then returns a response 311 to inform that the create-key-information command 310 has completed successfully. Then, the key generator 106 sends the reconstruct-key command 312 which stores the encryption key 109 encrypted by the temporary key in its data 207 to the receiving device 128. The key generator 125 uses the temporary key to decrypt the data 207 of the reconstruct-key command 312 received, and obtains the encryption key 126. Then, the key generator 125 returns a response 313 to inform that the reconstruct-key command 312 has completed successfully. Since the encryption keys 109 and 126 are encrypted and decrypted using the same temporary key, they are the same keys. The encryption key 126 is output from the key generator 125 to the decrypter 121. This completes the procedure for granting decrypting information.

The key generator 106 which has sent the reconstruct key command 312 outputs the output command 105 for encrypted output to the encrypter 101. The encrypter 101 receiving this command encrypts TSP received from the signal source 100 by the encryption key 109, and starts to output it to the source packet generator 102. This enables the sending device 110 to send the isochronous packet containing TSP encrypted by the encryption key 109 in its isochronous payload 952 on the 1394 bus 111. This isochronous packet received by the receiving device 128 is decrypted by the decrypter 121 using the encryption key 126 as described above, and outputs the decrypted packet to the AV generator 120.

In the above series of AKE control commands, each set of the make-response command 302 and verify-me command 304; make-response command 306 and verify-me command 308; and create-key-information command 310 and

reconstruct-key command 312, respectively has the same label 202. The make-response command 302, verify-me command 304, make-response command 306, verify-me command 308, create-key-information command 310, and reconstruct-key command 312 also have values 1, 2, 3, 4, 5, and 6 in the step No. 203 respectively.

If the actual data in the isochronous packet output from the sending device 110 changes from encrypted actual data to non-encrypted actual data, the decrypter 121 detects the change in the encrypting information 910, stops decryption, and outputs the data received from the actual data extractor 122 as it is to the AV generator 120.

If a bus reset occurs in the 1394 bus 111 after the aforementioned processes shown in Fig. 4 starts, the procedures after and make-response command 302 need to be repeated.

As described above, in the preferred embodiment of the present invention, the sending device sends encrypting information which indicates the encryption status of the actual data in the isochronous packet together with the actual data. This enables the receiving device receiving the isochronous packet to make a request to the sending device for an encryption key for decrypting the actual data if the receiving device detects, by checking the encrypting information in the isochronous packet, that the actual data is encrypted. The sending device receiving the request then gives the encryption key to the receiving device. Accordingly, the data transfer method of the present invention offers extremely simple procedures for giving the encryption key for decryption from the sending device to receiving device.

Moreover, in the preferred embodiment of the present invention, the isochronous packet transferred via isochronous communication contains i) encrypting information indicating the encryption status of the actual data and ii) actual data, but only the actual data is encrypted for data transfer. This makes possible a data transfer method which has no risk of erroneous operation when a conventional receiving device receives encrypted actual data while maintaining compatibility with the conventional data transfer method.

Furthermore, in the preferred embodiment of the present invention, the CIP header remains non-encrypted for transfer even if encryption of synchronous data starts while the receiving device is continuously receiving synchronous data sent by the sending device. This enables a data transfer method which eliminates the possibility of erroneous operation of the receiving device receiving the data.

In the preferred embodiment, once encryption by the encryption key starts, actual data in all transfer units is encrypted and sent. However, it is not necessary to encrypt all units of data to be transferred. For example, even if both encrypted transfer units and non-encrypted transfer units are sent alternately, the receiving device can correctly decrypt the data because encrypting information is included in the CIP header, thus achieving the same effect. In addition, it is apparent that the same effect is also achievable even if the receiving device specifies a percentage of encrypted transfer units to the sending device. The size of the MPEG source packet is 192 bytes, with more than one source packet stored in one isochronous payload in the case of the high data rate transfer of MPEG (12 Mbps minimum). Naturally, however, it is not possible to have both

encrypted source packet and non-encrypted source packet in the same isochronous payload.

In the preferred embodiment, all actual data is encrypted using the encryption key. However, it is not necessary to encrypt all pieces of data. For example, the same effect is achievable by encrypting the first half of the actual data, or encrypting the first and third quarters of the actual data. In this case, the receiving device can decrypt appropriately, if, when sending the data, information is inserted to indicate encrypted portions and their percentage in the CIP header. The same effect is also achievable by inserting in the CIP header encrypting information announcing whether the actual data is encrypted or not. The receiving device queries the sending device via asynchronous communication about which part of the actual data is encrypted and to what level, when the receiving device detects encryption by checking the CIP header. The same effect is also achievable in this case even when the receiving device specifies the encryption area and percentage to the sending device via asynchronous communication. If only the confidential portion in the actual data is encrypted, the burden for encryption and decryption is reduced, and at the same time, a sufficient effect of encryption may be achieved.

In the preferred embodiment, the isochronous packet containing only the CIP header without actual data is transferred until the completion of mutual authentication between the sending and receiving devices. However, the same effect is achievable even when an isochronous packet containing encrypted actual data is output from the start, and not the isochronous packet containing only the CIP header.

In the preferred embodiment, procedures for transferring the AKE control commands between sending and receiving devices are determined by mutual negotiation. However, if the receiving device features only one executable procedure, the same effect is achievable by starting to transfer commands immediately, without executing this negotiation procedure, using only the executable procedure. In this case, it may be preferable to specify in advance a basic minimum of executable procedures for all authorized devices.

In this preferred embodiment, direct authentication is implemented between the sending and receiving devices, following which decrypting information is transferred using a secret key. However, the means for transferring authentication and decrypting information is not limited to this procedure. For example, a public key may be used for mutual indirect authentication and the creation of a temporary key. Decrypting information may then be transmitted using this temporary key. Such procedures are briefly described below.

The sending and receiving device stores the key information necessary for mutual indirect authentication in the data 207 of the AKE control commands, and send this information to each other in line with a procedure determined by mutual negotiation. Here, the subfunction 299 is set to the exchange. This enables both sending and receiving devices to share the same temporary key if they are both authorized devices. Decrypting information is then transferred using the create-key-information command and reconstruct-key command in accordance with the same procedures as those described in the preferred embodiment.

In the preferred embodiment, the procedure for transferring AKE control commands exchanged between sending and receiving devices is determined by mutual negotiation. If the types of procedures executable by the sending device are known in advance, the same effect is achievable by having the receiving device transfer commands using a procedure executable by the sending device without first executing this negotiation procedure.

In the preferred embodiment, procedures for transferring AKE control commands exchanged between sending and receiving devices are determined by mutual negotiation. However, the method for determining transfer procedures is not limited to this one. More specifically, if priority is given in advance to each of several transfer procedures, the receiving device may start to transfer using the procedure given the highest priority which is executable by itself. If the sending device cannot execute that procedure, the receiving device tries to transfer data by going down the list of procedures in order of priority until a procedure that is executable by both the sending and receiving device is found. The AKE control commands are then transferred using this procedure to achieve the same effect.

In the preferred embodiment, the sending device encrypts decrypting information which is used for decrypting actual data before transferring it to the receiving device. However, the way the receiving device obtains decrypting information is not limited to this procedure. In other words, the sending device may provide the receiving device sufficient information for obtaining decrypting information, without transferring encrypted decrypting information, and the receiving device may obtain decrypting information indirectly from this information. More specifically, the sending device transfers only the type of hash

function to the receiving device, and the receiving device obtains decrypting information using the received type of hash function to achieve the same effect.

The preferred embodiment described above comprises an example of the AKE command format. However, the AKE command format is not limited to this one. In other words, the AKE command format indicated in this embodiment is just one example of how the preferred embodiment may be realized. The same effect is achievable by using commands in a different format.

Industrial applicability

As described above, the present invention has the significant effect of realizing a data transfer method using extremely simple procedures for passing key information for decryption from the sending device to the receiving device. Encryption identification information indicating the encryption status of actual data in synchronous data is sent together with actual data. The receiving device receiving the synchronous data checks the encryption identification information in the synchronous data, and if it detects that the actual data is encrypted, the receiving device requests the sending device for decrypting information for decrypting the encrypted data. The sending device receiving this request gives the decrypting information to the receiving device.

The present invention has another significant effect of realizing a data transfer method which eliminates the possibility of erroneous operation of the receiving device even if conventional receiving device receives encrypted synchronous data, while maintaining compatibility with a conventional data transfer method. Synchronous data transferred through synchronous communication contains i) encryption identification information indicating

encryption status of the actual data and ii) actual data, but only the actual data is encrypted for data transfer.

The present invention has still another significant effect of realizing a data transfer method which eliminates the possibility of erroneous operation of the receiving device even if encryption of synchronous data starts while the receiving device continuously receives synchronous data sent from the sending device. Synchronous data transferred through synchronous communication contains i) encryption identification information indicating encryption status of the actual data and ii) actual data, but only the actual data is encrypted for data transfer. This enables to transfer the CIP header as it is without being encrypted.

The present invention has still another significant effect of realizing a data transfer method which always executes the most suitable procedure even when new and conventional devices share the same network. A procedure for transferring and receiving both authentication and decrypting information with good future extendibility are achievable by selecting a procedure for providing authentication information and decrypting information exchanged between the sending and receiving devices by negotiation between the sending and receiving devices. In other words, even if a new authentication method or decrypting information become available in the future, the most suitable procedure will remain selectable by negotiation between devices even if a device which can use the new procedure and a device which can use only conventional procedures share the same network, as long as the new device is back-compatible with older procedures.

The present invention has still another significant effect of realizing decryption even if software which has a low encryption/decryption processing

speed is used. The present invention allows the relative proportion of encrypted actual data and non-encrypted actual data to be varied. Accordingly, even if the receiving device has no exclusive hardware for high-speed data decryption, software can be used instead. More specifically, even if the receiving device has no hardware for decryption like PC, rapid processing is made possible by reducing the proportion of encrypted data in the file and thus shortening the time required for the decryption process.

The present invention has still another significant effect of realizing a data transfer method which uses the limited bus transfer band efficiently and significantly reduces the risk of unauthorized device receiving readable data. Unless the sending and receiving devices mutually authenticate that both are authorized devices, isochronous packets without actual data are output.

ABSTRACT OF THE DISCLOSURE

A data transfer method which eliminates erroneous operation of conventional devices not supporting encryption when copy-protected AV information is encrypted and sent on an IEEE 1394 bus. Synchronous data transferred through isochronous communication contains i) encryption identification information for indicating encryption of actual data and ii) actual data. Only the actual data is encrypted. Encryption identification information indicating encryption of actual data in synchronous data is sent together with actual data from the sending device. A receiving device detecting encryption of actual data from this encryption identification information requests decrypting information from the sending device. The receiving device decrypts the actual data using decrypting information received from the sending device according to this request.

MAT-V07839

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: T. Nishimura et al. : Art Unit:
Serial No.: To Be Assigned : Examiner:
Filed: Herewith :
FOR: DATA TRANSFER METHOD :

PRELIMINARY AMENDMENT

Assistant Commissioner for Patents
Washington, D.C. 20231

S I R :

Prior to examination, please amend the above-identified
application as follows:

IN THE DRAWINGS:

Please delete the last two sheets of figures, also labeled as
"Reference Numerals".

Please substitute originally filed Fig. 2 with the copy of Fig. 2
as attached hereto.

IN THE SPECIFICATION:

After the title and before the first paragraph, please insert
--THIS APPLICATION IS A U.S. NATIONAL PHASE APPLICATION OF
PCT INTERNATIONAL APPLICATION PCT/JP98/01837--.

Please enter the substitute specification as attached hereto. Also enclosed is marked-up copy of the substitute specification showing additions and deletions.

IN THE CLAIMS:

Please cancel claims 12 through 15.

Please amend the claims as follows:

1 1. (Amended) A method for transferring data on a bus
2 system [in which] using both isochronous communication and asynchronous
3 communication [are employed]; said isochronous communication is for any
4 device on the bus to receive synchronous data; said asynchronous
5 communication is for a predetermined device to receive asynchronous data;
6 said synchronous data [may] capable of containing actual data[;] and [said
7 synchronous data also contains] encryption identification information [at an
8 area other than said actual data; said encryption identification information
9 indicates the status of encryption of said] indicating encrypted actual data;
10 and encrypted actual data is decrypted using decrypting information obtained
11 through the following steps:

12 a) [a receiving device] receiving said synchronous data [makes
13 a] at a receiving device, and said receiving device via said asynchronous
14 communication requesting [for] decrypting information [of] for said actual
15 data [to] from a sending device sending said synchronous data [via said

16 asynchronous communication], if said encryption identification [information
17 indicates that said] indicates encrypted actual data [is encrypted];

18 b) [said sending device] receiving said request [sends] at said
19 sending device and said sending device sending one of:

20 i) encrypted decrypting information of said actual data; and

21 ii) [data required for obtaining said] decrypting information data
22 for obtaining said decrypting information,

23 to said receiving device via said asynchronous communication;

24 and

25 c) [said receiving device executes] executing at said receiving
26 device one of:

27 i) [taking out] extracting said decrypting information from said
28 encrypted decrypting information [when said receiving device receives said
29 encrypted decrypting information]; and

30 ii) obtaining said decrypting information using said [data for
31 obtaining said] decrypting information data [when said receiving device
32 receives said data for obtaining decrypting information].

1 2. (Amended) The method for transferring data as defined
2 in Claim 1, wherein a plurality of [types of] procedures are available between
3 the steps of detecting [encryption of said] encrypted actual data and obtaining

4 said decrypting information by said receiving device receiving said
5 synchronous data; and said receiving device executes the [next] following
6 steps for obtaining said decrypting information before requesting said
7 decrypting information:

8 i) querying said sending device of types of procedures
9 executable by said sending device before requesting said decrypting
10 information;

11 ii) selecting a procedure from those executable by both said
12 sending device and receiving device; and

13 iii) obtaining said decrypting information in accordance with
14 said selected procedure.

1 3. (Amended) The method for transferring data as defined
2 in Claim 2, wherein [said] a procedure is selected in accordance with a
3 predetermined priority when there are a plurality of procedures executable by
4 both of said sending device and said receiving device.

1 4. (Amended) The method for transferring data as defined
2 in Claim 1, wherein a plurality of [types of] procedures are available between
3 the steps of detecting [encryption of said] encrypted actual data and obtaining
4 [of] said decrypting information by said receiving device receiving said
5 synchronous data; and said receiving device executes the [next] following
6 steps for obtaining said decrypting information:

7 i) starting [said] a procedure selected from said plurality of
8 [types of] procedures in accordance with a predetermined priority;

9 ii) re-selecting [one of] said procedures one-by-one until [said] a
10 procedure executable by said sending device is found [when the procedure
11 selected by said receiving device is not executable by said sending device];
12 and

13 iii) obtaining said decrypting information in accordance with the
14 selected procedure [when a procedure] executable by said sending device [is
15 found].

1 5. (Amended) The method for transferring data as defined
2 in [one of] Claim[s] 2 [to 4], wherein said asynchronous data transmitted
3 between said sending device and said receiving device in accordance with
4 said selected procedure contains an identifier for indicating the type of said
5 procedure executed.

1 6. (Amended) The method for transferring data as defined
2 in one of Claims 1 to 5, 16 and 17, wherein said receiving device
3 authenticates whether said sending device is an authorized sending device
4 before making a request for said decrypting information.

1 7. (Amended) The method for transferring data as defined
2 in one of Claims 1 to 5, 16 and 17, wherein said sending device receiving a
3 request for said decrypting information authenticates that said receiving

4 device is an authorized receiving device before sending encrypted decrypting
5 information of said actual data [after confirming].

1 8. (Amended) The method for transferring data as defined
2 in one of Claims 1 to 5, 16 and 17, wherein said sending device and said
3 receiving device [mutually] are authenticated [that both are] as authorized
4 [sending] devices [and receiving device] before said receiving device makes a
5 request for said decrypting information.

1 9. (Amended) The method for transferring data as defined
2 in one of Claims 1 to 5, 16 and 17 [8], wherein the [next] following steps are
3 executed before said receiving device makes a request for said decrypting
4 information:

5 i) said receiving device [sends] sending information required by
6 said sending device at least for [creating] establishing a common key [to]
7 with said sending device; and

8 ii) said sending device [sends] sending information required by
9 said receiving device at least for [creating] establishing said common key [to]
10 with said receiving device;

11 and [then] said sending device [encrypts] encrypting said
12 decrypting information using said common key and [sends] sending said
13 encrypted decrypting information; and said receiving device [takes out]
14 extracting said decrypting information from said encrypted decrypting
15 information received using said [common] encryption key.

1 11. (Amended) The method for transferring data as defined
2 in one of Claims 1 to 5, 16 and 17, wherein said sending device [has]
3 includes a signal source [of] for said actual data [inside] and determines
4 encryption of [each of] said actual data in a fixed length unit which is output
5 from said signal source; and said sending device places encrypted actual data
6 and non-encrypted actual data in different output units of said synchronous
7 communication, and then outputs them to said bus system.

 Please add new claims 16 through 20.

1 16. (Newly Added) The method for transferring data as
2 defined in Claim 3, wherein said asynchronous data transmitted between said
3 sending device and said receiving device in accordance with said selected
4 procedure contains an identifier for indicating the type of said procedure
5 executed.

1 17. (Newly Added) The method for transferring data as
2 defined in Claim 4, wherein said asynchronous data transmitted between said
3 sending device and said receiving device in accordance with said selected
4 procedure contains an identifier for indicating the type of said procedure
5 executed.

1 18. (Newly Added) The method for transferring data as
2 defined in Claims 1 to 5, 16 and 17, wherein said receiving device

3 authenticates whether said sending device is an authorized sending device
4 before making a request for said decrypting information.

1 19. (Newly Added) The method for transferring data as
2 defined in Claim 10, wherein the following steps are executed before said
3 receiving device makes a request for said decrypting information:

4 i) said receiving device sending information required by said
5 sending device at least for establishing a common key with said sending
6 device; and

7 ii) said sending device sending information required by said
8 receiving device at least for establishing said common key with said receiving
9 device;

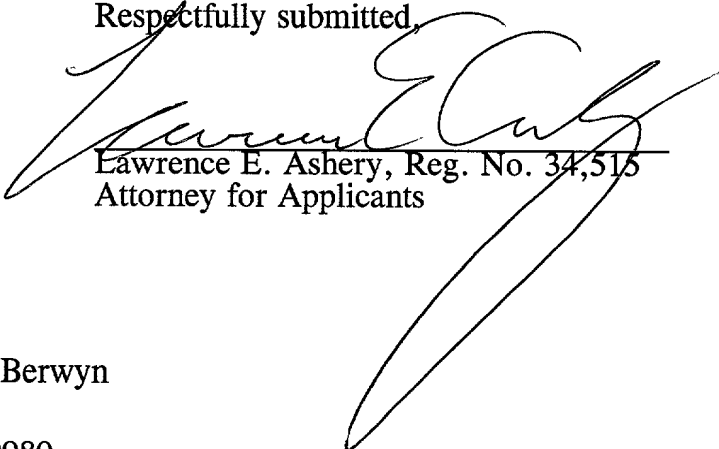
10 and said sending device encrypting said decrypting information
11 using said common key and sending said encrypted decrypting information;
12 and said receiving device extracting said decrypting information from said
13 encrypted decrypting information received using said common encryption
14 key.

1 20. (Newly Added) The method for transferring data as
2 defined in Claim 11, wherein the following steps are executed before said
3 receiving device makes a request for said decrypting information:

4 i) said receiving device sending information required by said
5 sending device at least for establishing a common key with said sending
6 device; and

7 ii) said sending device sending information required by said
8 receiving device at least for establishing said common key with said receiving
9 device;
10 and said sending device encrypting said decrypting information
11 using said common key and sending said encrypted decrypting information;
12 and said receiving device extracting said decrypting information from said
13 encrypted decrypting information received using said common encryption
14 key.

Respectfully submitted,



Lawrence E. Ashery, Reg. No. 34,515
Attorney for Applicants

LEA/lrs

Dated: October 25, 1999

Suite 301, One Westlakes, Berwyn
P.O. Box 980
Valley Forge, PA 19482-0980
(610) 407-0700

The Assistant Commissioner for Patents is
hereby authorized to charge payment to
Deposit Account No. 18-0350 of any fees
associated with this communication.

EXPRESS MAIL Mailing Label Number: EJ914196616US

Date of Deposit: October 25, 1999

I hereby certify that this paper and fee are being deposited, under 37 C.F.R. § 1.10 and with sufficient postage, using the "Express Mail Post Office to Addressee" service of the United States Postal Service on the date indicated above and that the deposit is addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231.



Kathleen Libby

DATA TRANSFER METHOD**FIELD OF THE INVENTION**

The present invention relates to the field of digital data transfer methods,
5 more particularly to the transfer of data in which normal digital data and encrypted digital data co-exist in the same data.

BACKGROUND OF THE INVENTION

One conventional data transfer method adopts the IEEE1394 standard
10 (IEEE: The Institute of Electrical and Electronics Engineers, Inc.). (Reference: IEEE Std 1394: 1995, High Performance Serial Bus.) In data transfer specified by the IEEE 1394 standard, there are two methods of communication. One is isochronous communication, which is suitable for transferring synchronous data such as digital video signals and digital audio signals. The other is asynchronous communication, which is suitable for
15 transferring asynchronous data such as control signals. Both methods of communication are applicable on the IEEE 1394 bus network. Isochronous communication is what is called "Broadcast communication, and an isochronous packet output from one device coupled to the IEEE 1394 bus is receivable by all the other devices coupled to the same bus. On the other hand, asynchronous communication is applicable to both one-to-one
20 communication and one-to-N broadcast communication. Each asynchronous packet output from one device coupled to the bus contains an identifier specifying the device(s) to which that packet is addressed. If this identifier specifies a particular device, only the device specified by the identifier receives the asynchronous packet. If the identifier specifies broadcast, all the devices coupled to the same bus receive the asynchronous
25 packet.

At present, the IEC (International Electrotechnical Commission) is preparing to stipulate the IEC1883 standard (hereafter referred to as AV protocol) for

transferring digital audio signals and digital video signals or transmitting data between devices coupled to an IEEE 1394 bus, employing the data transfer method conforming to the IEEE 1394 standard. In the AV protocol, video and audio data is located in the isochronous packet as shown in Fig. 5 and transferred. The isochronous packet includes a CIP (Common Isochronous Packet) header. The CIP header carries information that includes the type of AV data, the identification number of the device which is sending the isochronous packet, and the like.

Fig. 5 shows the format of the isochronous packet used in the AV protocol. The isochronous packet comprises an isochronous packet header 900, header CRC 901, isochronous payload 902, and data CRC 903. The isochronous packet header 900 contains a tag 907. The tag 907 shows that the isochronous packet conforms to the AV protocol when its value is 1. When the value of the tag 907 is 1, which means that the isochronous packet conforms to the AV protocol, the isochronous payload 902 has a CIP header 904 at its beginning. The CIP header 904 comprises a source ID 906 which identifies the device transmitting the isochronous packet. The CIP header 904 also comprises FMT 908 and FDF 909 which specify the type of actual data 905 in the isochronous payload 902. Digital AV data is contained in the actual data 905, but the actual data 905 is not always contained in the isochronous payload 902. Some packets may have an isochronous payload 902 which contains only the CIP header 904 without the actual data 905.

There is a group of commands called the AV/C Command Set for controlling devices in accordance with the AV protocol (Reference: 1394 TRADE ASSOCIATION Specification for AV/C Digital Interface Command Set Version 1.0, September 13, 1996). These commands and their responses are transferred by means of asynchronous communication.

In the conventional data transfer method as described above, compatibility with conventional devices which are not designed for transferring an encrypted

isochronous payload 902 cannot be secured when an encrypted isochronous packet, which contains the isochronous payload 902 which has been encrypted for copy protection, is sent. More specifically, conventional devices are designed with the precondition that the CIP header 904 is normally positioned at the beginning of the isochronous payload 902.

5 Accordingly, if the isochronous payload 902 is encrypted, conventional devices cannot correctly read out the encrypted CIP header 904, and decide that the isochronous packet does not conform to the AV protocol. A device receiving encrypted isochronous packets thus may not operate properly. In other words, such receiving devices cannot determine the type of data contained in the actual data 905, resulting in an inability to identify the
10 device transmitting the isochronous packet. In addition, asynchronous communication such as queries to the sending device are disabled. Accordingly, normal receiving operations cannot be carried out.

Furthermore, if the isochronous packet output from the sending device is encrypted while the receiving device is receiving the data, some conventional devices
15 may not be able to correctly read out the CIP header 904 as soon as encryption starts, resulting in inability to receive data properly.

In order to send AV information encrypted for copy protection from the sending device and decrypt the encrypted AV data by the authorized receiving device, the sending device needs to give decrypting information to the authorized receiving device. In
20 the conventional data transfer method, however, the sending device may be required to execute extremely complicated procedures in order to specify the receiving device. More specifically, each isochronous packet contains the source ID 906 which is the identifier of the sending device, but these packets do not contain information that identifies which device is authorized to receive these packets. The sending device thus cannot check which
25 device is receiving the isochronous packets during transmission of the isochronous packets. In order to find which of the devices coupled to the IEEE 1394 bus is receiving

the data, the sending device may require to query the data receiving status of every device coupled to the same bus. This makes the procedures for giving key information for decryption extremely complicated.

5

SUMMARY OF THE INVENTION

A data transfer method of the present invention satisfies the conventional communication standard even in the case of sending encrypted video and audio information via isochronous communication. In addition, the present invention offers a data transfer method which prevents erroneous operation even if conventional receiving
10 devices receive isochronous packets containing encrypted video and audio data.

The present invention still further offers a data transfer method which significantly simplifies procedures for giving key information for decryption from a sending device to an authorized receiving device.

In a data transfer method of the present invention, synchronous data
15 transferred via isochronous communication contains i) encryption identification information which indicates encryption status of actual data and ii) actual data, and only the actual data is encrypted.

To solve another problem in the conventional data transfer method, the encryption identification information which indicates encryption status of the actual data
20 in the synchronous data is sent together with the actual data from the sending device so that receiving device can detect that the actual data is encrypted based on this encryption identification information and request decrypting information to the sending device in the data transfer method of the present invention. Then, the receiving device receiving the decrypting information sent from the sending device upon request decrypts the actual data
25 using this decrypting information to complete data transfer.

Also in the data transfer method of the present invention, the receiving device receiving synchronous data checks for the encryption identification information contained in the synchronous data. If the receiving device detects that the actual data is encrypted, the receiving device requests for decrypting information for decrypting the actual data to the sending device. This request is made using a command in the AV/C set via asynchronous communication. At receiving this request, the sending device checks the packet header of received command to identify the device making the request, i.e., the receiving device. The sending device then gives decrypting information to the identified receiving device using a command via asynchronous communication, enabling to realize the data transfer method with extremely simple procedures for giving decrypting information from the sending device to receiving device.

Moreover, in the data transfer method of the present invention, only the actual data in the synchronous data is encrypted, and the encryption identification information indicating the encryption status of the actual data is included in the synchronous data. This enables to transfer the CIP header without being encrypted, preventing erroneous operation when the conventional device receives such encrypted synchronous data. In other words, the present invention realizes a data transfer method which assures compatibility with the conventional data transfer method and eliminates the possibility of erroneous operation when the conventional receiving device receives encrypted synchronous data.

Furthermore, the data transfer method of the present invention eliminates the possibility of erroneous operation of the receiving device receiving data when encryption of synchronous data starts while continuously receiving synchronous data from the sending device because the CIP header is not encrypted and transferred as it is.

BRIEF DESCRIPTION OF THE DRAWING

Fig. 1 is a schematic view of a format of a CIP header in accordance with a preferred embodiment of the present invention.

Fig. 2 is a block diagram illustrating functions of sending and receiving devices in accordance with the preferred embodiment of the present invention.

5 Fig. 3A is a format of AKE status command in accordance with the preferred embodiment of the present invention.

Fig. 3B is a format of AKE response to the AKE status command in accordance with the preferred embodiment of the present invention.

10 Fig. 3C is a format of AKE control command in accordance with the preferred embodiment of the present invention.

Fig. 4 is a schematic view illustrating procedures for transmitting an asynchronous packet between sending and receiving devices in accordance with the preferred embodiment of the present invention.

15 Fig. 5 is a format of isochronous packet in a data transfer method of the prior art.

DESCRIPTION OF THE PREFERRED EMBODIMENT

A preferred embodiment of the present invention is described next with reference to drawings.

20 Fig. 1 shows a format of the payload of an isochronous packet to be transferred in the preferred embodiment of the present invention. The preferred embodiment is one example of the transfer of a TSP (Transport Packet) in accordance with MPEG (the Moving Picture Expert Group) specifications. The ENC (hereafter referred to as encryption status) 910 indicates whether the actual data 905 is encrypted or
25 not.

Fig. 2 shows the relation between sending and receiving devices in the preferred embodiment of the present invention. A sending device 110 and receiving device 128 are coupled via an IEEE 1394 bus (hereafter referred to as a 1394 bus) 111.

First, the functions of each block in the sending device 110 are described.

5 A signal source 100 outputs an MPEG transport packet TSP (not illustrated) in an 188 byte unit, which will be sent via the 1394 bus 111, to an encrypter 101. In other words, in the preferred embodiment, the signal source 100 outputs data with a fixed length of 188 bytes. The encrypter 101 encrypts and outputs the TSP received from the signal source 100 using an encryption key 109 provided by a key generator 106. In the
10 preferred embodiment, the encryption key is equivalent to the decrypting information. An output command 105 is a command from the key generator 106 to the encrypter 101. There are three types of commands: normal output, encrypted output, and empty output. If the encrypter 101 receives the output command 105 for normal output, the TSP received from the signal source 100 is output without modification, and registers the value 0 as the
15 encrypting information 910. If the output command 105 is for encrypted output, the encrypter 101 encrypts the TSP with the encryption key 109 received from the key generator 106, and registers the value 1 as the encrypting information 910. If the output command 105 is for empty output, the encrypter 101 outputs an empty signal (not illustrated) every time it receives a TSP from the signal source 100, and registers the
20 value 1 as the encrypting information 910. A source packet generator 102 adds a 4-byte source packet header to the 188-byte TSP received from the encrypter 101, and outputs a 192-byte source packet (actual data 905). A CIP block generator 103 adds a CIP header 954 to the source packet received from the source packet generator 102, and outputs an isochronous payload 952. Here, the CIP block generator 103 places the encrypting
25 information 910 received from the encrypter 101 in the CIP header 954. An isochronous packet generator 107 adds an isochronous packet header 900, header CRC 901, and data

CRC 903 to the isochronous payload 952 received from the CIP block generator 103, and outputs an isochronous packet. Since the content of the isochronous payload 952 is data that conforms to the AV protocol, the value of the tag 907 is set to 1. The key generator 106 sends the encryption key 109 to the receiving device 128 by communicating the asynchronous packet with the receiving device 128, as shown in Fig. 3, which is described later. The key generator 106 also outputs the encryption key 109 to the encrypter 101 as described above.

10 A 1394 packet I/O means 108 inputs and outputs isochronous and asynchronous packets between the 1394 bus 111 and sending device 110. More specifically, the 1394 packet I/O means 108 outputs the isochronous packet received from the isochronous packet generator 107 and asynchronous packet received from the key generator 106 to the 1394 bus 111, and also outputs asynchronous packet received from the 1394 bus 111 to the key generator 106.

Next, functions of each block of the receiving device 128 are described.

15 A 1394 packet I/O means 127 inputs and outputs isochronous and asynchronous packets between the 1394 bus 111 and receiving device 128. More specifically, the 1394 packet I/O means 127 outputs the isochronous packet received from the 1394 bus 111 to a payload extractor 123, and outputs asynchronous packet received from the 1394 bus 111 to a key generator 125. The 1394 packet I/O means 127 also
20 outputs asynchronous packet received from the key generator 125 to the 1394 bus 111.

The payload extractor 123 receives the isochronous packet, transmitted from the 1394 bus 111, from the 1394 packet I/O means 127. When the value of the isochronous packet tag 907 is 1, the payload extractor 123 determines that an isochronous payload 952 contains data conforming to the AV protocol, and outputs the isochronous
25 payload 952 to an actual data extractor 122. When received isochronous payload 952 contains the actual data 905, the actual data extractor 122 outputs the actual data 905 to a

decrypter 121, after removing the CIP header 954 placed at the beginning of the isochronous payload 952. The actual data extractor 122 also outputs the source ID 906 and encrypting information 910 extracted from the CIP header 954 to the key generator 125. The encrypting information 910 is also output to the decrypter 121. The key
5 generator 125 receives an encryption key 126 as a result of exchanging asynchronous packet with the sending device 110 via asynchronous communication, which is described later, and outputs the encryption key 126 to the decrypter 121. When the value of the encrypting information 910 received from the actual data extractor 122 is 0, the decrypter 121 outputs the actual data 905 received from the actual data extractor 122 to an AV
10 generator 120 as it is. When the value of the encrypting information 910 is 1, the decrypter 121 decrypts the actual data 905 using the encryption key 126 received from the key generator 125, and outputs decrypted actual data 905 to the AV generator 120.

Next, the transmission of an asynchronous packet via the aforementioned asynchronous communication setup is described.

15 Figs. 3A to 3C illustrate how the format of the asynchronous packet is transmitted by asynchronous communication. More specifically, Figs. 3A and 3C show the command formats of the AKE commands (AKE: Authentication and Key Exchange) communicated between the key generators 106 and 125. Fig. 3B shows the response format. These commands and responses belong to the AV/C Command Set, and are
20 communicated between the sending device 110 and receiving device 128 using the asynchronous communication. By communicating these commands and responses, the sending device 110 and receiving device 128 exchange information required for the authentication of each other and encryption keys 109 and 126. The AKE commands comprise AKE control commands for requesting a target device to carry out a specific
25 operation, and an AKE status commands for querying the status and capabilities of the target device.

Fig. 3A shows the format of the AKE status command. In the AKE status command, an operation code 208 indicates that this command is an AKE command. The value of the algorithm ID 200 is set at 0, with other values reserved for future extension.

Fig. 3B shows the format of responses to the AKE status commands. This is a response sent back from the device receiving the AKE status command to the device issuing the AKE status command. There are multiple procedures for exchanging information for mutual authentication and transmission of encryption keys 109 and 126 between the sending device 110 and receiving device 128. In an algorithm field 201, the identifier for an information exchange procedure which the device returning an applicable response can execute is assigned in bits. In other words, the receiving device 128 exchanges several commands and responses with the sending device 110 after an encrypted TSP is detected in line with the aforementioned procedures and before receiving the encryption keys 109 and 126. There is more than one procedure for communicating these commands and responses. The device sending back the response designates the executable information exchange procedure by setting 1 to an applicable bit in the algorithm field 201. Since the size of the algorithm field 201 is 16 bits, a maximum of 16 types of information exchange procedures can be indicated. The maximum data length 212 indicates the longest receivable data length in the form of bytes for exchanging AKE commands and responses.

Fig. 3C shows the format of the AKE control commands. The algorithm field 201 in the AKE control commands set informs of an executed information exchange procedure when the value of the algorithm ID 200 is 0. Only one bit in the algorithm field 201 of the AKE control command and the response to AKE control commands is set at 1, and the other bits are 0. A bit having the value 1 indicates the information exchange procedure being used. A label 202 is used for identifying correspondence between AKE control commands. For example, let's say a certain information exchange procedure

specifies that the device receiving an AKE control command needs to return a different AKE control command corresponding to the AKE control command received when the AKE control command is sent from one device to another. In this case, the label 202 inserted in the returned AKE control command will have the same value as the label 202 inserted in the first AKE control command received, in order to clarify the correlation between both AKE control commands. In step No. 203, a serial number from 1 is given to each AKE control command in the sequence of communication in the information exchange procedure.

A subfunction 299 takes the values shown in Table 1, and the meaning of each AKE command is determined by these values.

Table 1

Subfunction	Value
Make-response	0016
Verify-me	0116
Create-key-	1016
information Reconstruct-key	1116
Exchange	2016

If the subfunction 299 is the make-response, this AKE control command challenges the authentication of the device receiving this command. Here, the data 207 contains authentication challenge data expressed as random numbers to authenticate the receiving device. The device receiving this command returns an AKE control command whose subfunction 299 is set to verify-me.

When returning the AKE control command, the data stored in the data 207 is the authentication response data which is a result of a predetermined operation with respect to the authentication challenge data in the received data 207. The key information used for this operation is a key given only to an authorized device in advance. Whether

the device executing the operation is an authorized device or not can be determined by checking the returned authentication response data.

If the subfunction 299 is the create-key-information, this AKE control command requests the encryption key 109 to the device receiving this command. The device receiving this AKE control command returns the AKE control command whose subfunction 299 is set to reconstruct-key. At this point, the encrypted encryption key 109 is stored in the data 207 and returned.

If the subfunction 299 is the exchange, this AKE control command requests the exchange of key information between devices sending and receiving the command. This key information is stored in the data 207 and transferred for indirect authentication between devices or the creation of a common key.

Values other than those specified in Table 1 are reserved for future extension. The channel No. 204 indicates the channel number for isochronous communication between the sending device 110 and receiving device 128. This channel No. 204 is valid only when the subfunction 299 is set to the create-key-information or reconstruct-key. In other cases, this value will be set to FF in hexadecimal format. Block No. 205 and total block No. 206 are used when data which should be handled by the AKE control command cannot be sent by one AKE command. In this case, applicable data is divided into blocks, and transferred in several transmissions. The total block No. 206 indicates the number of divided blocks in applicable data. The block No. 205 indicates the number of each block in the data 207. The data length 209 indicates the valid data length, as bytes, in the data 207. The data 207 is data exchanged by the AKE control command. The device receiving the AKE control command returns a response to that specific AKE control command. The format and value of the response are the same as those of the received AKE control command. The only detail which differs is that the response does not contain the data 207.

Fig. 4 shows a chronological example of AV/C commands which are exchanged between the sending device 110 and receiving device 128 before sending the encryption keys 109 and 126 from the sending device 110 to receiving device 128. First, operations of both devices before exchanging AV/C commands shown in Fig. 4 are briefly described.

An initial condition is that non-encrypted TSP is sent from the sending device 110. The TSP output from the signal source 100 is input to the encrypter 101. Since the output command 105 is set to the normal output, the encrypter outputs TSP as it is without encryption to the source packet generator 102, and registers the value 0 as the encrypting information 910. The source packet generator 102 adds 4-byte source packet header to the TSP received, and outputs it to the CIP block generator 103. The CIP block generator 103 adds 8-byte CIP header 954, and outputs it as isochronous payload 952 to the isochronous packet generator 107. Here, the encrypting information 910 contained in the CIP header 954 is 0 which is input from the encrypter 101. The isochronous packet generator adds the isochronous packet header 900, header CRC 901, and data CRC 903 to the received isochronous payload 952 to create the isochronous packet. This isochronous packet is output to the 1394 bus 111 by the 1394 packet I/O means 108. Since the applicable isochronous packet conforms to the AV protocol, the tag 907 in the isochronous packet header 900 is set to 1.

When the TSP output from the signal source 100 is changed, which means that AV information changes from that unprotected AV information to copy-protected AV information, the key generator 106 detects this change, and changes the output command 105 from the normal output to empty output. At the same time, the encryption key 109 for encrypting TSP is given to the encrypter 101.

When the output command 105 is for empty output, the encrypter 101 outputs an empty signal to the source packet generator 102 every time it receives a TSP

from the signal source 100, and registers the value 1 as the encrypting information. At receiving the empty signal from the encrypter 101, the source packet generator 102 transmits the received empty signal as it is to the CIP block generator 103 without adding the source packet header. When the CIP block generator 103 receives the empty signal, it
5 outputs only the CIP header 954 to the isochronous packet generator 107. Here, the encrypting information 910 in the CIP header 954 uses the value 1 output from the encrypter 101. The isochronous packet generator 107 creates an isochronous packet as the isochronous payload 952 using the CIP header 954 received from the CIP block generator, and outputs it to the 1394 packet I/O means 108. Since this isochronous packet conforms
10 to the AV protocol, the value of the tag 907 is set to 1. The 1394 packet I/O means 108 outputs received isochronous packet to the 1394 bus 111. This isochronous packet is continuously output, and the isochronous packet only containing the CIP header 954 in this isochronous payload 952 is continuously output to the 1394 bus 111. The receiving device 128 receiving this isochronous packet checks its tag 907 by the 1394 packet I/O
15 means 127, detects that the isochronous packet conforms to the AV protocol, and then outputs this isochronous packet to the payload extractor 123. The payload extractor 123 extracts the isochronous payload 952 from received isochronous packet, and outputs it to the actual data extractor 122. The actual data extractor 122 outputs the encrypting information 910 and source ID 906 in the CIP header 954 to the key generator 125. After
20 the key generator 125 detects that the value of the encrypting information 910 is 1, the receiving device 128 learns that device outputting the isochronous packet from the source ID 906 is the sending device 110. Then, the key generator 125 finally goes onto a process for requesting the encryption keys 109 and 126 using the A/C commands, as shown in Fig. 4.

25 In Fig. 4, the AKE status command 300 is first sent from the receiving device 128 to sending device 110. This enables the receiving device 128 to query

information exchange procedure that can be used by the sending device 110. Replying to this query, the sending device 110 returns the AKE response 301 to the receiving device 128. Information exchange procedure which the sending device 110 can execute is assigned in bits in the algorithm field 201 of the AKE response 301. This allows the receiving device 128 to learn which information exchange procedures can be executed by the sending device. For example, if the sending device 110 can execute the second and sixth information exchange procedures, binary indication in the algorithm field 201 of the AKE response 301 will be 0000000000100010.

The receiving device 128 receiving the AKE response 301 selects one optimal procedure from information exchange procedures that both sending device 110 and receiving device can execute. Then, AV/C commands are exchanged according to the selected exchange procedure. Let's say the receiving device 128 can execute the second and eighth information exchange procedures. Then, the information exchange procedure which can be executed by both sending device 110 and receiving device 128 is only the second procedure. Accordingly, the rest of authentication and information exchange are executed using the second procedure. In the AKE control command in this procedure, the value of algorithm ID will be 0 and the value of the algorithm field 201 will be 0000000000000010 in hexadecimal indication. The information exchange procedure specifies not only the sequence of exchanging a range of AKE control commands but also a format and processing method of the data 207 sent by each AKE control command.

In accordance with the second information exchange procedure, the key generator 125 sends the make response command 302 to the sending device 110. In the data 207 of this make response command 302, two random numbers RRa and RRb generated by the key generator 125 are encrypted, and the algorithm field 201 contains identification information indicating the use of the second procedure. The key used for encryption is a common secret key given to both authorized sending device and receiving

device in advance. The key generator 106 receiving the make-response command 302 checks the algorithm field 201 of the received make-response command 302, and learns to use the second procedure for the rest of authentication and information exchange. Since the key generator 106 can execute the second procedure, the key generator 106 knows that the data 207 of the make response command 302 sent in accordance with this second procedure contains two random numbers encrypted by this secret key. After taking out two random numbers RRa and RRb from the data 207 using this secret key, the key generator 106 returns a response 303 to inform that a response can be generated. Then, the key generator 106 stores one of the random numbers RRa taken out in the data 207, and sends the verify-me command 304 to the receiving device 128. This is the response requested by the previous make-response command 302. Hereafter, the algorithm field 201 of each AKE command exchanged between the sending device 110 and receiving device 128 always contain the identification information indicating the second procedure.

The key generator 125 receiving the verify-me command 304 confirms that RRa in the data 207 conforms to the random number RRa generated by itself, and then returns a response 305 to the verify-me command 304 to inform that verification has completed successfully. The key generator 125 then finally authenticates that the sending device 110 is an authorized sending device.

The sending device 110 then use the make-response command 306 and verify-me command 308 in accordance with the procedures after the make-response command 302 described above to confirm that the receiving device 128 is an authorized receiving device. However, the random number used here is RTa and RTb, and the random number sent back by the verify-me command 308 is RTb.

Now that both sending device 110 and receiving device 128 know the random numbers RRb and RTb, and have confirmed that both are authorized devices, the key generator 106 and key generator 125 separately generates a temporary key (not

illustrated) from RRB and RTb using a common operation method specified by the second procedure. These temporary keys are a common key only between the sending device 110 and receiving device 128.

Next, the key generator 125 sends the create-key-information command 310 to the sending device 110. A channel number of the isochronous packet that the receiving device 128 is currently receiving is stored in the channel No. 204 of the create-key-information command 310. The key generator 106 receiving this create-key-information command 310 encrypts the encryption key 109 to be used for encrypting TSP with the aforementioned temporary key, and then returns a response 311 to inform that the create-key-information command 310 has completed successfully. Then, the key generator 106 sends the reconstruct-key command 312 which stores the encryption key 109 encrypted by the temporary key in its data 207 to the receiving device 128. The key generator 125 uses the temporary key to decrypt the data 207 of the reconstruct-key command 312 received, and obtains the encryption key 126. Then, the key generator 125 returns a response 313 to inform that the reconstruct-key command 312 has completed successfully. Since the encryption keys 109 and 126 are encrypted and decrypted using the same temporary key, they are the same keys. The encryption key 126 is output from the key generator 125 to the decrypter 121. This completes the procedure for granting decrypting information.

The key generator 106 which has sent the reconstruct key command 312 outputs the output command 105 for encrypted output to the encrypter 101. The encrypter 101 receiving this command encrypts TSP received from the signal source 100 by the encryption key 109, and starts to output it to the source packet generator 102. This enables the sending device 110 to send the isochronous packet containing TSP encrypted by the encryption key 109 in its isochronous payload 952 on the 1349 bus 111. This isochronous packet received by the receiving device 128 is decrypted by the decrypter using the

encryption key 126 as described above, and outputs the decrypted packet to the AV generator 120.

In the above series of AKE control commands, each set of the make response command 302 and verify-me command 304; make-response command 306 and
5 verify-me command 308; and create-key-information command 310 and reconstruct-key command 312, respectively has the same label 202. The make-response command 302, verify me command 308, create-key-information command 310, and reconstruct key command 312 also have values 1, 2, 3, 4, 5, and 6 in the step No. 203 respectively.

If the actual data 105 in the isochronous packet output from the sending
10 device 110 changes from encrypted actual data 105 to non-encrypted actual data 105, the decrypter 121 detects the change in the encrypting information 910, stops decryption, and outputs the data received from the actual data extractor 122 as it is to the AV generator 120.

If a bus reset occurs in the 1394 bus 111 after the aforementioned processes
15 shown in Fig. 4 starts, the procedures after and make-response command 302 need to be repeated.

As described above, in the preferred embodiment of the present invention, the sending device sends encrypting information which indicates the encryption status of the actual data in the isochronous packet together with the actual data. This enables the
20 receiving device receiving the isochronous packet to make a request to the sending device for an encryption key for decrypting the actual data if the receiving device detects, by checking the encrypting information in the isochronous packet, that the data is encrypted. The sending device receiving the request then gives the encryption key to the receiving device. Accordingly, the data transfer method of the present invention offers extremely
25 simple procedures for giving the encryption key from the sending device to receiving device for decryption.

Moreover, in the preferred embodiment of the present invention, the isochronous packet transferred via isochronous communication contains i) encrypting information indicating the encryption status of the actual data and ii) actual data, but only the actual data is encrypted for data transfer. This makes possible a data transfer method
5 which has no risk of erroneous operation when a conventional receiving device receives encrypted actual data while maintaining compatibility with the conventional data transfer method.

Furthermore, in the preferred embodiment of the present invention, the CIP header remains non-encrypted for transfer even if encryption of synchronous data starts
10 while the receiving device is continuously receiving synchronous data sent by the sending device. This enables a data transfer method which eliminates the possibility of erroneous operation of the receiving device receiving the data.

In the preferred embodiment, once encryption by the encryption key starts, actual data in all transfer units is encrypted and sent. However, it is not necessary to
15 encrypt all units of data to be transferred. For example, even if both encrypted transfer units and non-encrypted transfer units are sent alternately, the receiving device can correctly decrypt the data because encrypting information is included in the CIP header, thus achieving the same effect. In addition, it is apparent that the same effect is also achievable even if the receiving device specifies a percentage of encrypted transfer units
20 to the sending device. The size of the MPEG source packet is 192 bytes, with more than one source packet stored in one isochronous payload in the case of the high data rate transfer of MPEG (12 Mbps minimum). Naturally, however, it is not possible to have both encrypted source packet and non-encrypted source packet in the same isochronous payload.

25 In the preferred embodiment, all actual data is encrypted using the encryption key. However, it is not necessary to encrypt all pieces of data. For example,

the same effect is achievable by encrypting the first half of the actual data, or encrypting the first and third quarters of the actual data. In this case, the receiving device can decrypt appropriately, if, when sending the data, information is inserted to indicate encrypted portions and their percentage in the CIP header. The same effect is also achievable by inserting in the CIP header encrypting information announcing whether the actual data is encrypted or not. The receiving device queries the sending device via asynchronous communication about which part of the actual data is encrypted and to what level, when the receiving device detects encryption by checking the CIP header. The same effect is also achievable in this case even when the receiving device specifies the encryption area and percentage to the sending device via asynchronous communication. If only the confidential portion in the actual data is encrypted, the burden for encryption and decryption is reduced, and at the same time, a sufficient effect of encryption may be achieved.

In the preferred embodiment, the isochronous packet containing only the CIP header without actual data is transferred until the completion of mutual authentication between the sending and receiving devices. However, the same effect is achievable even when an isochronous packet containing encrypted actual data is output from the start, and not the isochronous packet containing only the CIP header.

In the preferred embodiment, procedures for transferring the AKE control commands between sending and receiving devices are determined by mutual negotiation. However, if the receiving device features only one executable procedure, the same effect is achievable by starting to transfer commands immediately, without executing this negotiation procedure, using only the executable procedure. In this case, it may be preferable to specify in advance a basic minimum of executable procedures for all authorized devices.

In this preferred embodiment, direct authentication is implemented between the sending and receiving devices, following which decrypting information is transferred using a secret key. However, the means for transferring authentication and decrypting information is not limited to this procedure. For example, a public key may be used for mutual indirect authentication and the creation of a temporary key. Decrypting information may then be transmitted using this temporary key. Such procedures are briefly described below.

The sending and receiving device stores the key information necessary for mutual indirect authentication in the data 207 of the AKE control commands, and send this information to each other in line with a procedure determined by mutual negotiation. Here, the subfunction 299 is set to the exchange. This enables both sending and receiving devices to share the same temporary key if they are both authorized devices. Decrypting information is then transferred using the create-key-information command and reconstruct-key command in accordance with the same procedures as those described in the preferred embodiment.

In the preferred embodiment, the procedure for transferring AKE control commands exchanged between sending and receiving devices is determined by mutual negotiation. If the types of procedures executable by the sending device are known in advance, the same effect is achievable by having the receiving device transfer commands using a procedure executable by the sending device without first executing this negotiation procedure.

In the preferred embodiment, procedures for transferring AKE control commands exchanged between sending and receiving devices are determined by mutual negotiation. However, the method for determining transfer procedures is not limited to this one. More specifically, if priority is given in advance to each of several transfer procedures, the receiving device may start to transfer using the procedure given the

highest priority which is executable by itself. If the sending device cannot execute that procedure, the receiving device tries to transfer data by going down the list of procedures in order of priority until a procedure that is executable by both the sending and receiving device is found. The AKE control commands are then transferred using this procedure to
5 achieve the same effect.

In the preferred embodiment, the sending device encrypts decrypting information which is used for decrypting actual data before transferring it to the receiving device. However, the way the receiving device obtains decrypting information is not limited to this procedure. In other words, the sending device may provide the receiving
10 device sufficient information for obtaining decrypting information, without transferring encrypted decrypting information, and the receiving device may obtain decrypting information indirectly from this information. More specifically, the sending device transfers only the type of hash function to the receiving device, and the receiving device obtains decrypting information using the received type of hash function to achieve the
15 same effect.

The preferred embodiment described above comprises an example of the AKE command format. However, the AKE command format is not limited to this one. In other words, the AKE command format indicated in this embodiment is just one example of how the preferred embodiment may be realized. The same effect is achievable by using
20 commands in a different format.

Industrial applicability

As described above, the present invention has the significant effect of realizing a data transfer method using extremely simple procedures for passing key
25 information from the sending device to the receiving device for decryption. Encryption identification information indicating the encryption status of actual data in synchronous

data is sent together with actual data. The receiving device receiving the synchronous data checks the encryption identification information in the synchronous data, and if it detects that the actual data is encrypted, the receiving device requests the sending device for decrypting information for decrypting the encrypted data. The sending device receiving
5 this request gives the decrypting information to the receiving device.

The present invention has another significant effect of realizing a data transfer method which eliminates the possibility of erroneous operation of the receiving device even if conventional receiving device receives encrypted synchronous data, while maintaining compatibility with a conventional data transfer method. Synchronous data
10 transferred through synchronous communication contains i) encryption identification information indicating encryption status of the actual data and ii) actual data, but only the actual data is encrypted for data transfer.

The present invention has still another significant effect of realizing a data transfer method which eliminates the possibility of erroneous operation of the receiving
15 device even if encryption of synchronous data starts while the receiving device continuously receives synchronous data sent from the sending device. Synchronous data transferred through synchronous communication contains i) encryption identification information indicating encryption status of the actual data and ii) actual data, but only the actual data is encrypted for data transfer. This enables to transfer the CIP header as it is
20 without being encrypted.

The present invention has still another significant effect of realizing a data transfer method which always executes the most suitable procedure even when new and conventional devices share the same network. A procedure for both authentication and decrypting information with good future extendibility are achievable by selecting a
25 procedure for providing authentication information and decrypting information exchanged between the sending and receiving devices by negotiation between the sending and

receiving devices. In other words, even if a new authentication method or decrypting information become available in the future, the most suitable procedure will remain selectable by negotiation between devices even if a device which can use the new procedure and a device which can use only conventional procedures share the same
5 network, as long as the new device is back-compatible with older procedures.

The present invention has still another significant effect of realizing decryption even if software which has a low encryption/decryption processing rate is used. The present invention allows the relative proportion of encrypted actual data and non-encrypted actual data to be varied within a single file. Accordingly, even if the
10 receiving device has no exclusive hardware for high-speed data decryption, software can be used instead. More specifically, even if the receiving device has no hardware for decryption like PC, rapid processing is made possible by reducing the proportion of encrypted data in the file and thus shortening the time required for the decryption process.

The present invention has still another significant effect of realizing a data
15 transfer method which uses the limited bus transfer band efficiently and significantly reduces the risk of unauthorized device receiving readable data. Unless the sending and receiving devices mutually authenticate that both are authorized devices, isochronous packets without actual data are output.

What is claimed is:

1 1. A method for transferring data on a bus system in which both isochronous
2 communication and asynchronous communication are employed; said isochronous
3 communication is for any device on the bus to receive synchronous data; said
4 asynchronous communication is for a predetermined device to receive asynchronous data;
5 said synchronous data may contain actual data; said synchronous data also contains
6 encryption identification information at an area other than said actual data; said
7 encryption identification information indicates the status of encryption of said actual data;
8 and encrypted actual data is decrypted using decrypting information obtained through the
9 following steps:

10 a) a receiving device receiving said synchronous data makes a request for
11 decrypting information of said actual data to a sending device sending said synchronous
12 data via said asynchronous communication, if said encryption identification information
13 indicates that said actual data is encrypted;

14 b) said sending device receiving said request sends one of:

15 i) encrypted decrypting information of said actual data; and

16 ii) data required for obtaining said decrypting information

17 to said receiving device via said asynchronous communication; and

18 c) said receiving device executes one of:

19 i) taking out said decrypting information from said encrypted decrypting
20 information when said receiving device receives said encrypted decrypting information;

21 and

22 ii) obtaining said decrypting information using said data for obtaining

23 said decrypting information when said receiving device receives said data for obtaining

24 decrypting information.

25

1 2. The method for transferring data as defined in Claim 1, wherein a
2 plurality of types of procedures are available between the steps of detecting encryption of
3 said actual data and obtaining said decrypting information by said receiving device
4 receiving said synchronous data; and said receiving device executes the next steps for
5 obtaining said decrypting information before requesting said decrypting information:

6 i) querying said sending device of types of procedures executable by said
7 sending device;

8 ii) selecting a procedure from those executable by both sending device and
9 receiving device; and

10 iii) obtaining said decrypting information in accordance with said selected
11 procedure.

12

1 3. The method for transferring data as defined in Claim 2, wherein said
2 procedure is selected in accordance with a predetermined priority when there are a
3 plurality of procedures executable by both of said sending device and said receiving
4 device.

5

1 4. The method for transferring data as defined in Claim 1, wherein a
2 plurality of types of procedures are available between the steps of detecting encryption of
3 said actual data and obtaining of said decrypting information by said receiving device
4 receiving said synchronous data; and said receiving device executes the next steps for
5 obtaining said decrypting information:

6 i) starting said procedure selected from said plurality of types of procedures
7 in accordance with a predetermined priority;

8 ii) re-selecting one of said procedures until said procedure executable by
9 said sending device is found when the procedure selected by said receiving device is not
10 executable by said sending device; and

11 iii) obtaining said decrypting information in accordance with the selected
12 procedure when a procedure executable by said sending device is found.

13
1 5. The method for transferring data as defined in one of Claims 2 to 4,
2 wherein said asynchronous data transmitted between said sending device and said
3 receiving device in accordance with said selected procedure contains an identifier for
4 indicating the type of said procedure executed.

5
1 6. The method for transferring data as defined in one of Claims 1 to 5,
2 wherein said receiving device authenticates whether said sending device is an authorized
3 sending device before making a request for said decrypting information.

4
1 7. The method for transferring data as defined in one of Claims 1 to 5,
2 wherein said sending device receiving a request for said decrypting information
3 authenticates that said receiving device is an authorized receiving device before sending
4 encrypted decrypting information of said actual data after confirming.

5
1 8. The method for transferring data as defined in one of Claims 1 to 5, said
2 sending device and said receiving device mutually authenticate that both are authorized
3 sending device and receiving device before said receiving device makes a request for said
4 decrypting information.

5

1 9. The method for transferring data as defined in one of Claims 1 to 8,
2 wherein the next steps are executed before said receiving device makes a request for said
3 decrypting information:

4 i) said receiving device sends information required by said sending device at
5 least for creating a common key to said sending device; and

6 ii) said sending device sends information required by said receiving device
7 at least for creating said common key to said receiving device;

8 and then said sending device encrypts decrypting information using said
9 common key and sends said encrypted decrypting information; and said receiving device
10 takes out said decrypting information from said encrypted decrypting information
11 received using said common key.

12
1 10. The method for transferring data as defined in one of Claims 1 to 5,
2 wherein only said actual data is encrypted.

3
1 11. The method for transferring data as defined in one of Claims 1 to 5,
2 wherein said sending device has a signal source of said actual data inside and determines
3 encryption of each of said actual data in a fixed length unit output from said signal source;
4 and said sending device places encrypted actual data and non-encrypted actual data in
5 different output units of said synchronous communication, and then outputs them to said
6 bus system.

7
1 12. The method for transferring data as defined in Claim 11, wherein said
2 receiving device specifies a percentage of said encrypted actual data and said non-
3 encrypted actual data to said sending device using said asynchronous communication; and

4 said sending device changes the percentage of encryption in accordance with said
5 specification.

6

1 13. The method for transferring data as defined in one of Claims 1 to 5,
2 wherein said sending device has a signal source of said actual data inside, and determines
3 a percentage of encryption of said actual data in a fixed length unit output from said
4 signal source; and said sending device places said actual data in an output unit of said
5 synchronous communication, and then outputs it to said bus system.

6

1 14. The method for transferring data as defined in Claim 13, wherein said
2 receiving device specifies a percentage of said encryption to said sending device using
3 said asynchronous communication, and said sending device changes the percentage of
4 encryption in accordance with said specification.

5

1 15. The method for transferring data as defined in one of Claims 1 to 5,
2 wherein said sending device sends said synchronous data excluding said actual data until
3 at least said decrypting information is requested; and said sending device starts sending
4 synchronous data containing said actual data only after at least receiving said request for
5 said decrypting information.

6

ABSTRACT OF THE DISCLOSURE

A data transfer method which eliminates erroneous operation of conventional devices not supporting encryption when copy-protected AV information is encrypted and sent on the IEEE 1394 bus. Synchronous data transferred through isochronous communication contains i) encryption identification information for indicating encryption of actual data and ii) actual data. Only the actual data is encrypted. Encryption identification information indicating encryption status of actual data in synchronous data is sent together with actual data from the sending device. A receiving device detecting encryption of actual data from this encryption identification information requests for decrypting information to the sending device. The receiving device decrypts the actual data using decrypting information received from the sending device according to this request.

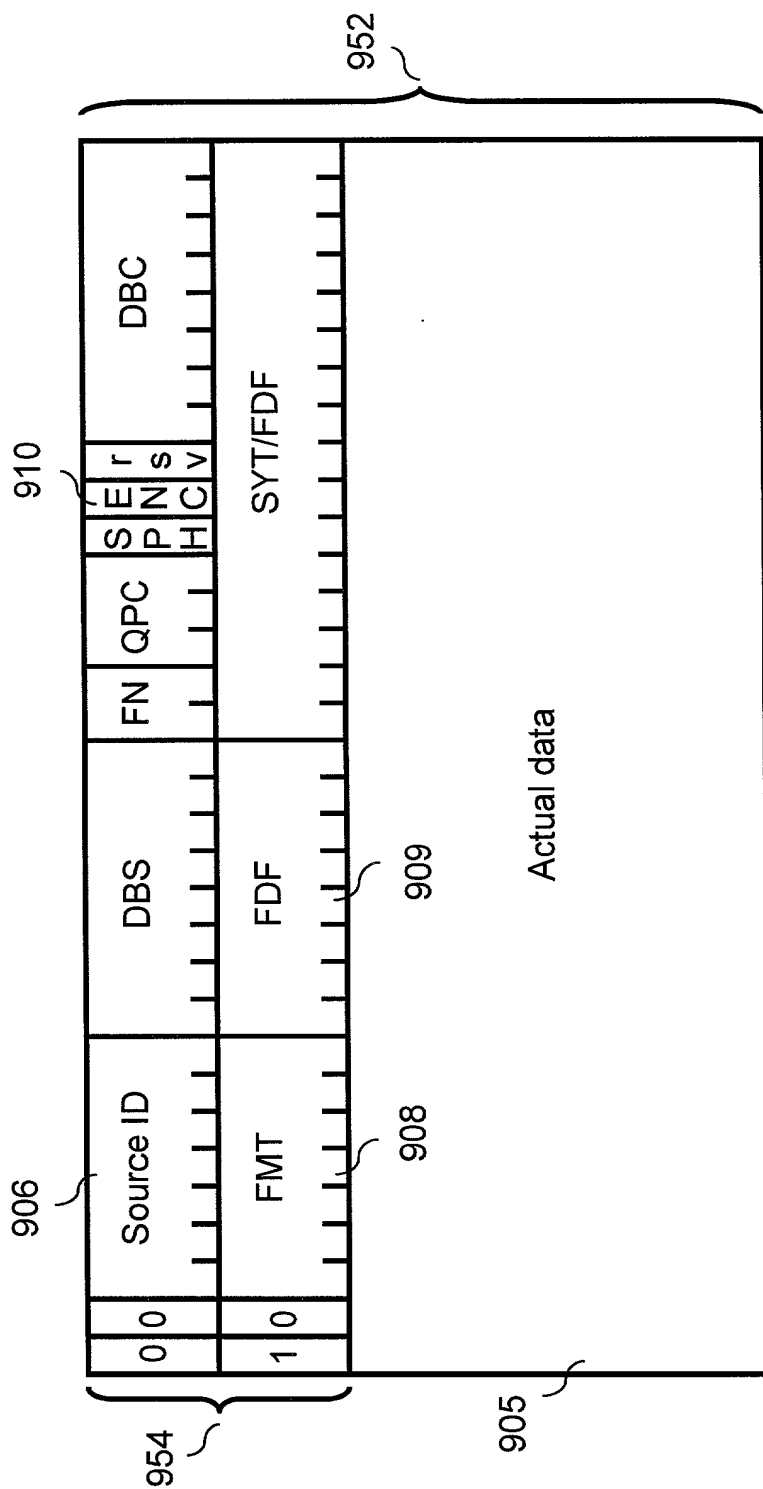


FIG. 1

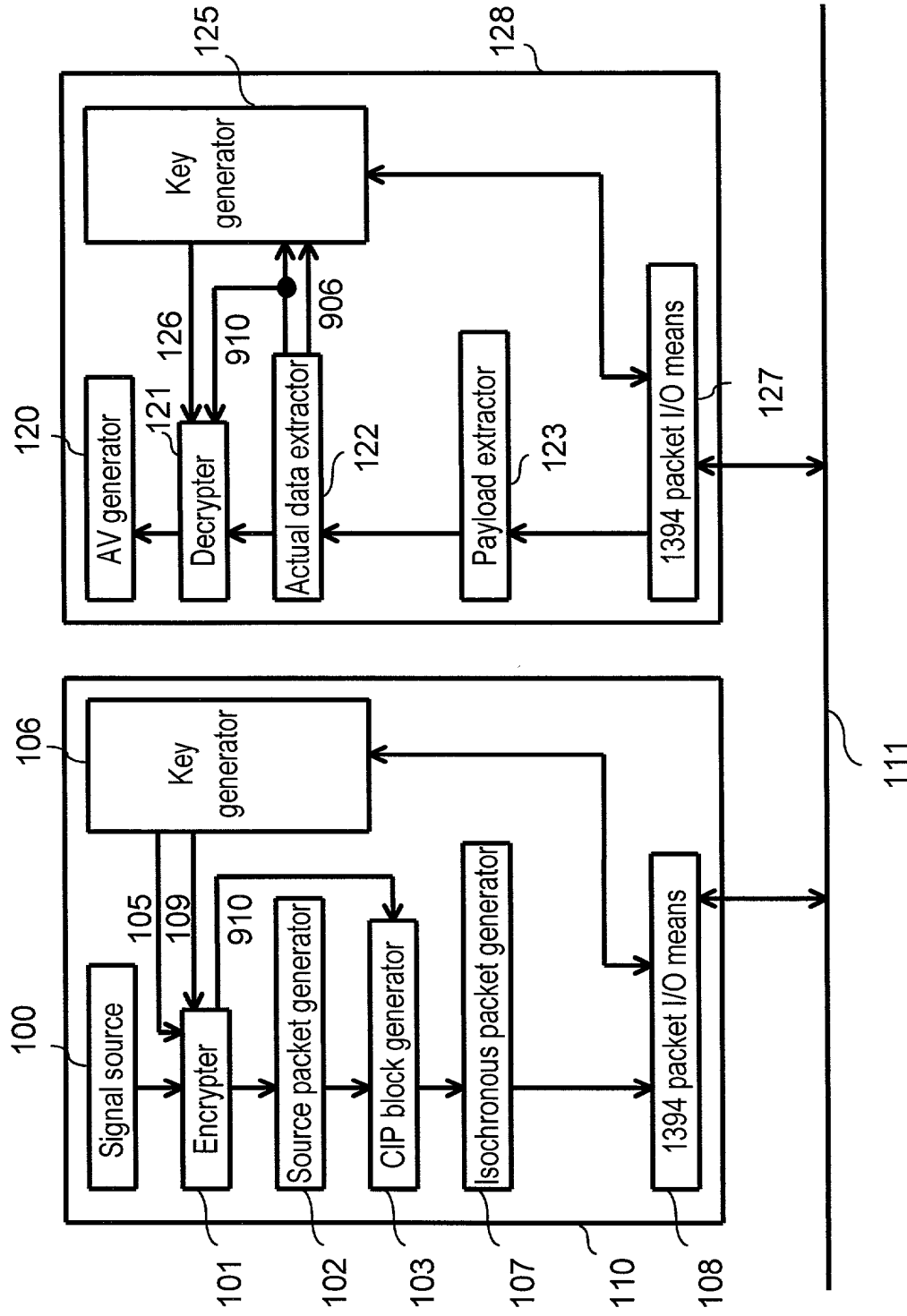


FIG. 2

FIG. 3A

opcode	Authentication and Key exchange																msb	lsb
operand[0]	F ₁₅								algorithm ID									
operand[1]	FF ₁₆																	
operand[2]	FF ₁₆																	
operand[3]	FF ₁₆																	
operand[4]	FF ₁₆																	
operand[5]	FF ₁₆																	
operand[6]	FF ₁₆																	
operand[7]	FF ₁₆																	
operand[8]	FF ₁₆																	

FIG. 3B

	msb						lsb
opcode	Authentication and Key exchange						
operand[0]	0			algorithm ID			
operand[1]	(msb) algorithm field (lsb)						
operand[2]							
operand[3]	FF ₁₆						
operand[4]	FF ₁₆						
operand[5]	FF ₁₆						
operand[6]	FF ₁₆						
operand[7]	(msb) maximum data length (lsb)						
operand[8]							

FIG. 3C

	msb					lsb					
opcode	Authentication and Key exchange										
operand[0]	reserved			algorithm ID			200				
operand[1]	algorithm field						201				
operand[2]							(msb)				(lsb)
operand[3]							label 202		step No.		
operand[4]	subfunction						299				
operand[5]	channel No.						204				
operand[6]	block No. 205		total block No.				206				
operand[7]	data_length						209				
operand[8]							(msb)				(lsb)
operand[9]											
operand[8+ data_length]	data						207				

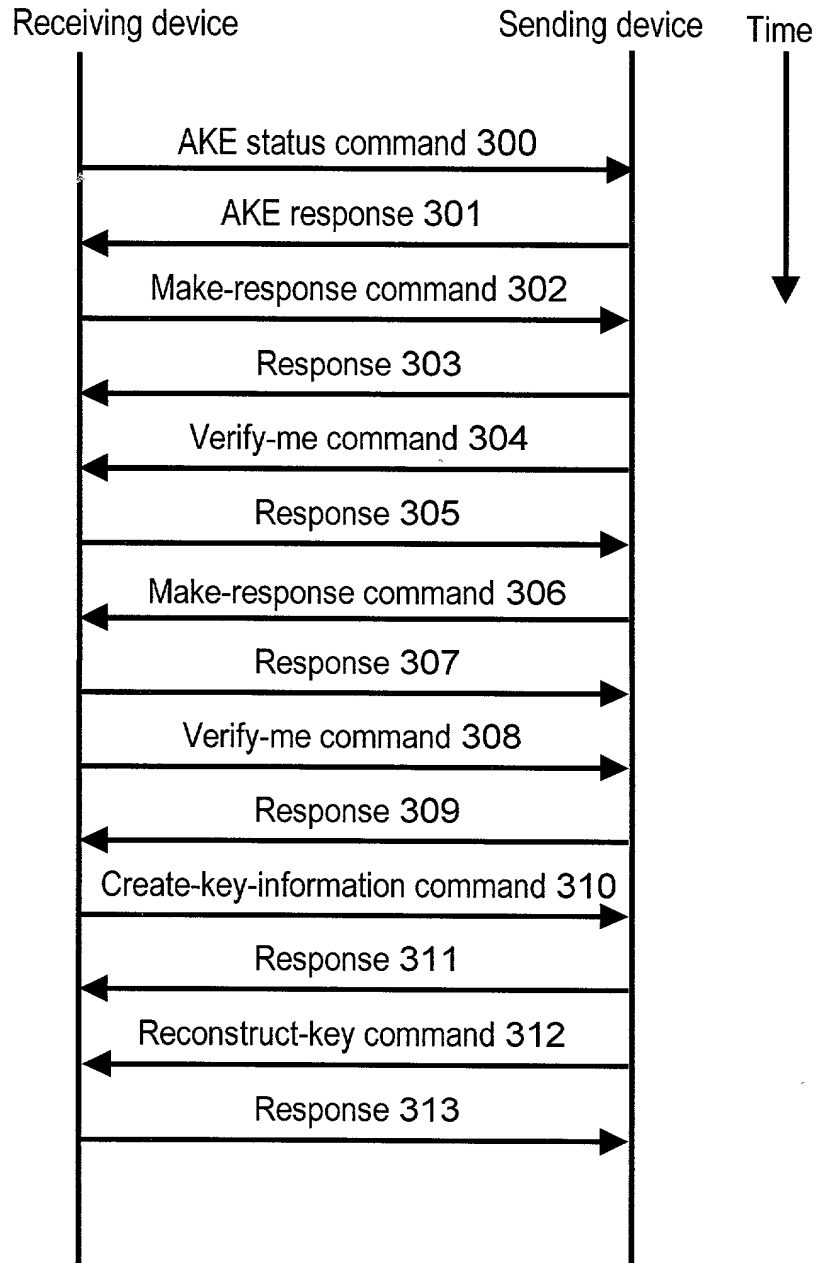


FIG. 4

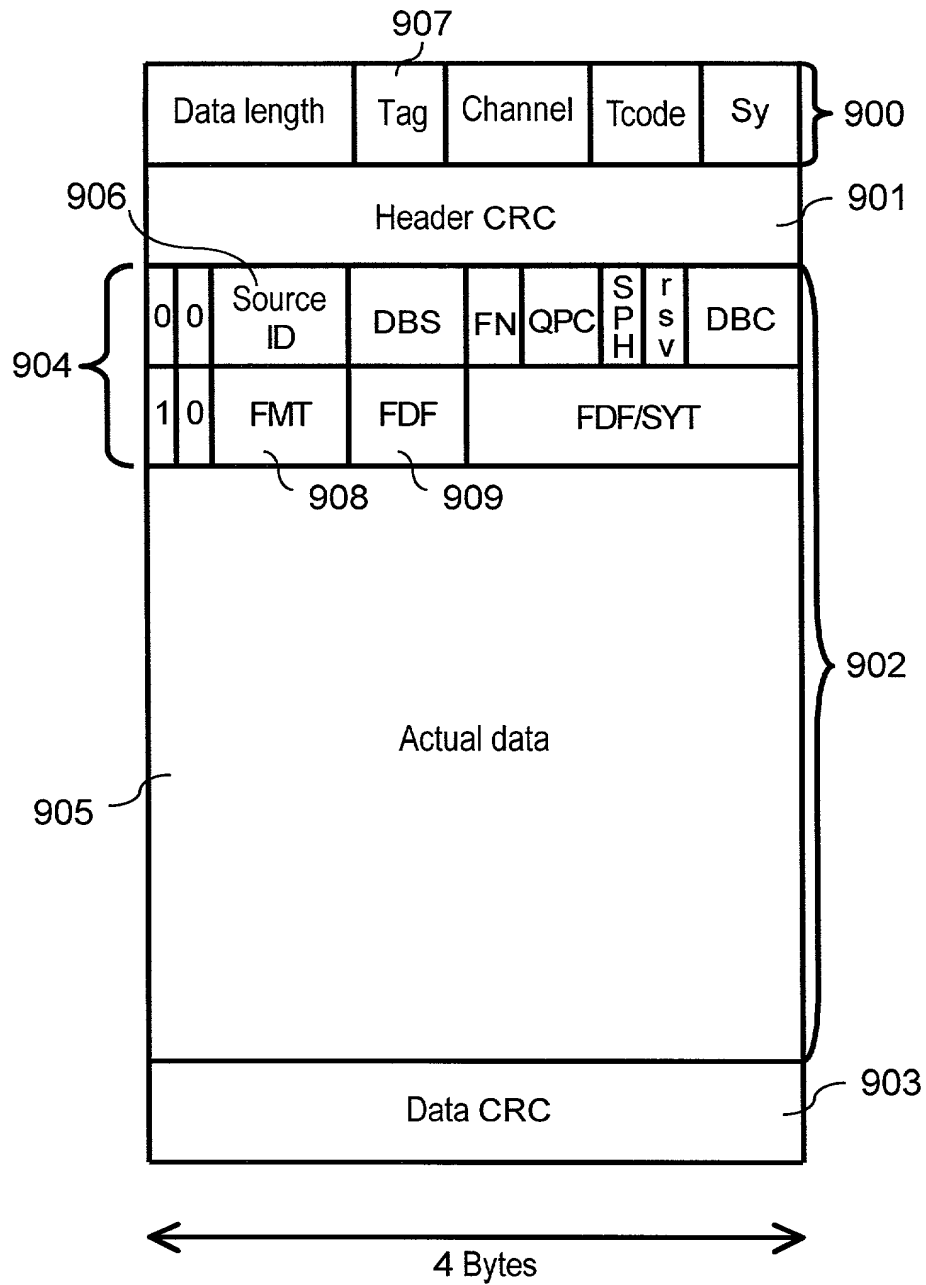


FIG. 5 PRIOR ART

Reference numerals

	100	signal source
	101	encrypter
	102	source packet generator
5	103	CIP block generator
	107	isochronous packet generator
	108, 127	1394 packet I/O means
	105	output command
	109, 126	encryption key
10	110	sending device
	128	receiving device
	111	IEEE 1394 bus
	106, 125	key generator
	120	AV generator
15	121	decrypter
	122	actual data extractor
	123	payload extractor
	200	algorithm ID
	201	algorithm field
20	202	label
	203	step No.
	204	channel No.
	205	block No.
	206	total block No.
25	207	data
	208	operation code

	209	data length
	212	maximum data length
	299	subfunction
	300	AKE status command
5	301	AKE response
	302, 306	make-response command
	303, 305, 307, 309, 311, 313	response
	304, 308	verify-me command
10	310	create-key-information command
	312	reconstruct-key command
	900	isochronous packet header
	901	header CRC
	902, 952	isochronous payload
15	903	data CRC
	904, 954	CIP header
	905	actual data
	906	source ID
	907	tag
20	908	FMT
	909	FDF
	910	encrypting information (ENC)
	952	isochronous payload

Declaration and Power of Attorney For Patent Application

English Language Declaration

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name,

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

DATA TRANSFER METHOD,

the specification of which is attached hereto unless the following box is checked:



was filed on April 22, 1998 as

United States Application Number or PCT International Application Number PCT/JP98/01837

and was amended on (if applicable).

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR § 1.56.

I hereby claim foreign priority benefits under 35 U.S.C. §119(a)-(d) or § 365(b) of any foreign application(s) for patent or inventor's certificate, or § 365(a) of any PCT International application which designated at least one country other than the United States, listed below and have also identified below by checking the box, any foreign application for patent or inventor's certificate, or PCT International application having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)

Priority Not Claimed

9-106995

Japan

24 April 1997

(Number)

(Country)

(Day/Month/Year Filed)

☐

(Number)

(Country)

(Day/Month/Year Filed)

☐

I hereby claim the benefit under 35 U.S.C. § 119(e) of any United States provisional application(s) listed below.

(Application Number)

(Filing Date)

(Application Number)

(Filing Date)

I hereby claim the benefit under 35 U.S.C. § 120 of any United States application(s), or 365(c) of any PCT International application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of 35 U.S.C. § 112, I acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR § 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application:

(Application Number) (Filing Date) (Status - patented, pending, abandoned)

(Application Number) (Filing Date) (Status - patented, pending, abandoned)

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith:

Paul F. Prestia	Reg. No. <u>23,031</u>	Lawrence E. Ashery	Reg. No. <u>34,515</u>	Mark J. Marcelli	Reg. No. <u>36,593</u>
Allan Ratner	Reg. No. <u>19,717</u>	Christopher R. Lewis	Reg. No. <u>36,201</u>	Joshua L. Cohen	Reg. No. <u>38,040</u>
Andrew L. Ney	Reg. No. <u>20,300</u>	Robert L. Andersen	Reg. No. <u>25,771</u>	Christopher J. Dervishian	Reg. No. <u>42,480</u>
Kenneth N. Nigon	Reg. No. <u>31,549</u>	Daniel N. Calder	Reg. No. <u>27,424</u>	Jack J. Jankovitz	Reg. No. <u>42,690</u>
Kevin R. Casey	Reg. No. <u>32,117</u>	Louis W. Beardell, Jr.	Reg. No. <u>40,506</u>		
Benjamin E. Leace	Reg. No. <u>33,412</u>	Jacques L. Etkowicz	Reg. No. <u>41,738</u>		
James C. Simmons	Reg. No. <u>24,842</u>	Eric A. Dichter	Reg. No. <u>41,708</u>		

Address all correspondence to: Lawrence E. Ashery

Ratner & Prestia, Suite 301, One Westlakes, Berwyn, P.O. Box 980, Valley Forge, PA 19482-0980

Address all telephone calls to: Lawrence E. Ashery at (610) 407-0700.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full name of sole or first inventor (given name, family name) Takuya Nishimura

Inventor's signature Takuya Nishimura Date December 9, 1999

Residence Osaka, Japan

Citizenship Japanese

Post Office Address 6-1-105, Myokenzaka, Katano-shi, Osaka 576-0021 Japan

Full name of second joint inventor, if any (given name, family name) Hiroyuki Iitsuka

Second Inventor's signature Hiroyuki Iitsuka Date December 9, 1999

Residence Osaka, Japan

Citizenship Japanese

Post Office Address 6-25-6, Kisaichi, Katano-shi, Osaka, 576-0033 Japan



Additional inventors are being named on separately numbered sheets attached hereto.

Full name of third joint inventor, if any (given name, family name) Masazumi Yamada

3-10 Third inventor's signature Masazumi Yamada Date December 9, 1999

Residence Osaka, Japan

Citizenship Japanese

Post Office Address 6-24-10, Kinda-cho, Moriguchi-shi, Osaka 570-0011 Japan

Full name of fourth joint inventor, if any (given name, family name) ____

Fourth inventor's signature _____ Date _____

Residence ____

Citizenship ____

Post Office Address ____

Full name of fifth joint inventor, if any (given name, family name) ____

Fifth inventor's signature _____ Date _____

Residence ____

Citizenship ____

Post Office Address ____

Full name of sixth joint inventor, if any (given name, family name) ____

Sixth inventor's signature _____ Date _____

Residence ____

Citizenship ____

Post Office Address ____

Full name of seventh joint inventor, if any (given name, family name) ____

Seventh inventor's signature _____ Date _____

Residence ____

Citizenship ____

Post Office Address ____